October 1995
£2·45

ELEKTOR ELECTRONICS

THE INTERNATIONAL ELECTRONICS MAGAZINE

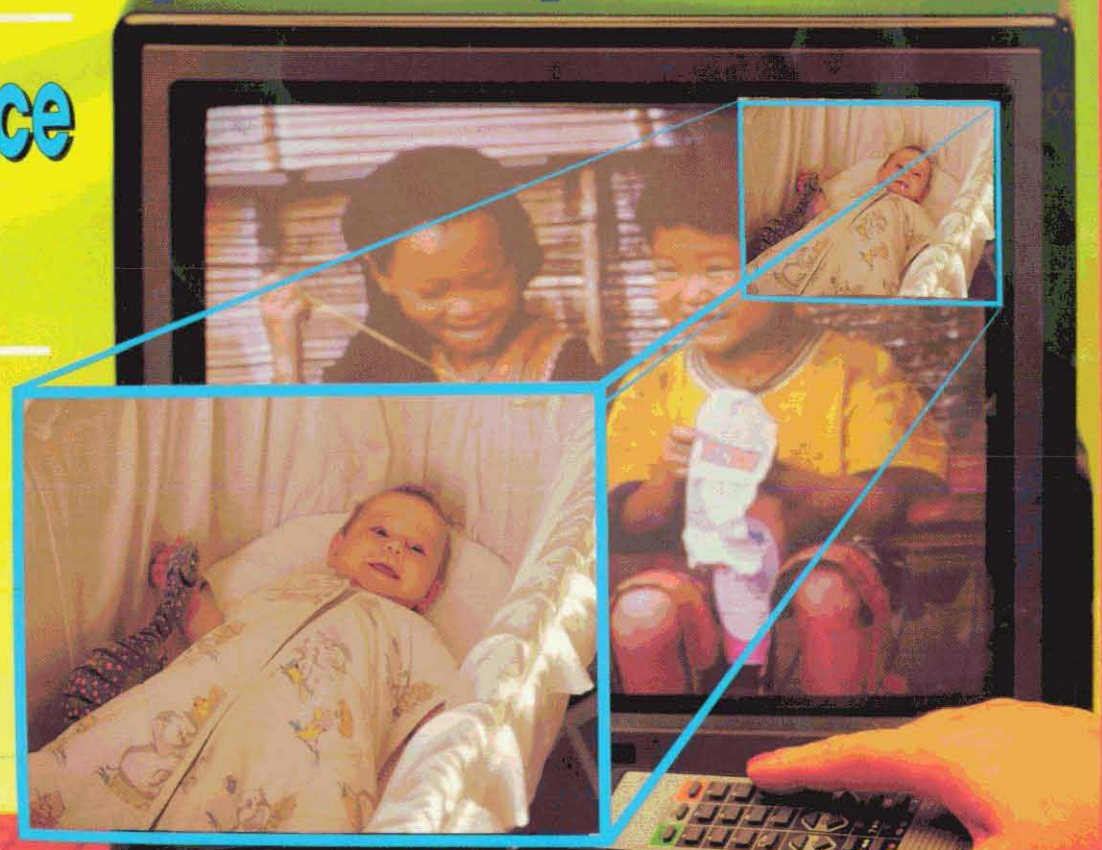
Focus on: Measuring with a PC

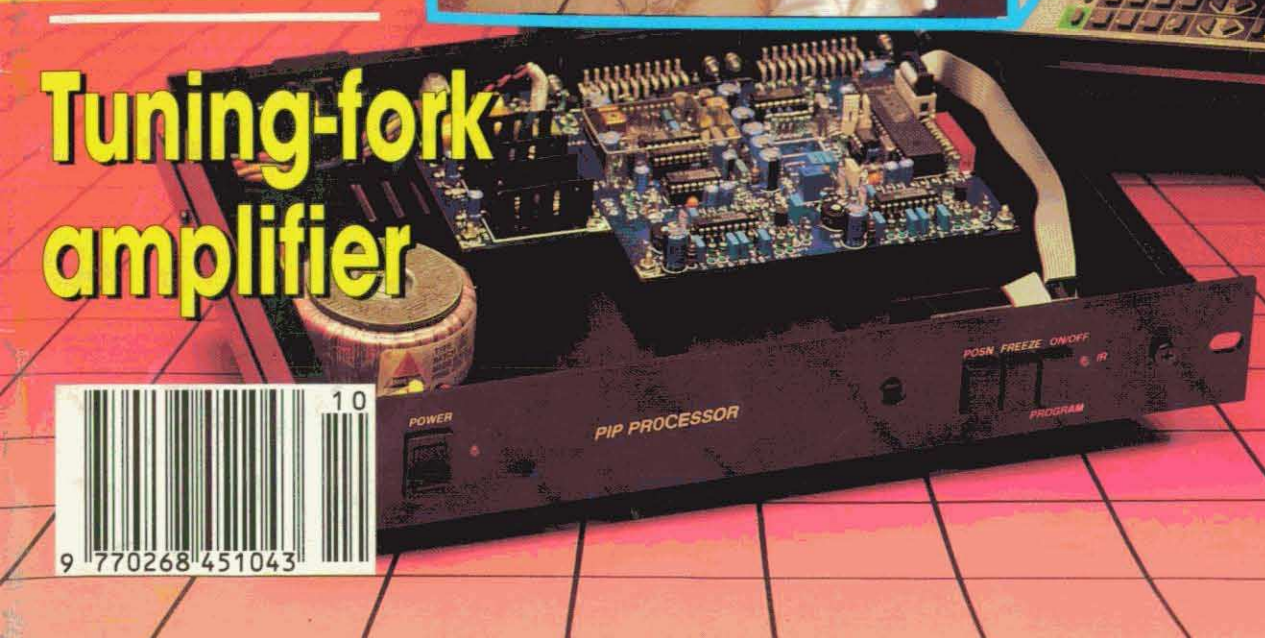
Picture-in-picture processor

Capacitance meter

Matchbox BASIC computer

Tuning-fork amplifier





9 770268 451043 10

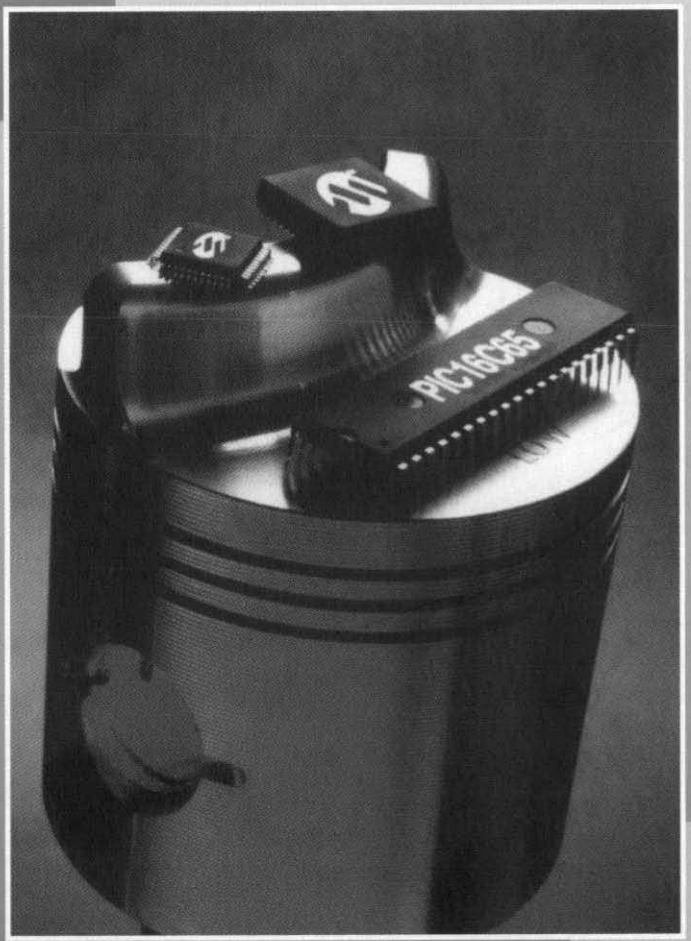
AUDIO & HI-FI • COMPUTERS & MICROPROCESSORS • DESIGN IDEAS • RADIO, TELEVISION & COMMUNICATIONS • SCIENCE & TECHNOLOGY • TEST & MEASUREMENT

ELEKTOR ELECTRONICS

Special Autumn Supplement

PLC Emulation using PIC™ Microcontrollers

by Walter Ditch

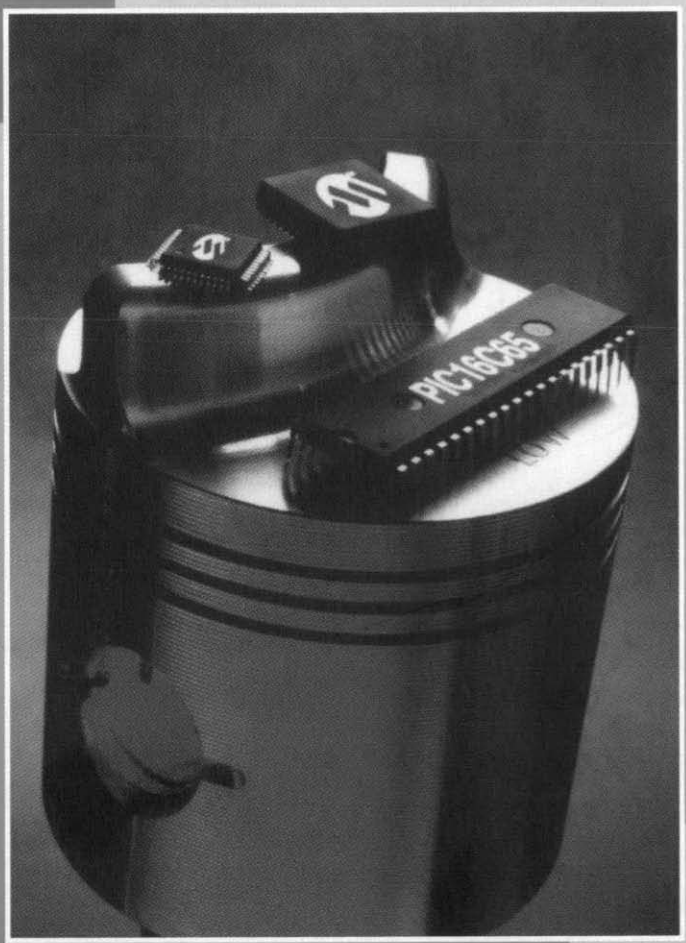


ELEKTOR ELECTRONICS

Special Autumn Supplement

PLC Emulation using PIC™ Microcontrollers

by Walter Ditch



Contents

1	Introduction	3
2	Instruction set	3
	Combinational logic	3
	Latches, counters and timers	9
	Shift registers and the drum sequencer	15
3	Method of operation	25
4	Testing your designs	26
	Using the assembler and simulator	26
	Creating your own programs	26
	Hardware design	27
	A sample application	29
5	Summary	30

The diskette for 'PLC emulation using PIC microcontrollers' is available through the Readers Services. It is supplied free of charge with every order for the 'Experimentation board for PICs', order code 944105-1 (July/August 1994). For price and ordering information, please refer to the Readers Services page in your most recent issue of *Elektor Electronics*. This offer is subject to stock availability, and available until February 1996.

Published by Elektor Electronics (Publishing), P.O. Box 1414, Dorchester DT2 8YH, England.

PLC Emulation using PIC™ Microcontrollers

by Walter Ditch

1. Introduction

THIS supplement shows how Microchip Technology's 16C54 and 16C55 microcontrollers may be programmed using *ladder logic*, which is traditionally associated with *programmable logic controllers* or *PLCs*. The task of developing control-oriented software is greatly simplified by this approach, when compared with assembly language programming.

Ladder logic owes its origin to the use of relays to implement combinational and sequential logic functions, particularly in industry. Ladder diagrams were used to specify the wiring of the relay circuitry, so a ladder diagram is really a specialized form of electrical circuit diagram. The difficulty of requiring and maintaining these complex wiring arrangements led directly to the adoption of PLCs, which use software rather than hardware to implement their control function.

As a general rule, any design problem which involves combinational logic, latches, counters and timers is likely to be suitable for solution using ladder logic. The fact that program design involves the interconnection of these electrical building blocks makes this approach to software development particularly suited to engineers with an electrical, rather than computing background.

All program listings and the PLC emulation software itself are available on diskette from the Readers Services. This disk comes free of charge with the PIC experimentation board described further on.

2. Instruction Set

Combinational Logic

As an example of the degree of simplification which can be achieved with ladder logic, recall the 'LED flasher' assembly program which was included in part 3 of the *PIC Programming Course* (*Elektor Electronics* October 1994). Even after removing the various comments and assembler directives (which were intended to explain the program's operation and encourage a structured programming technique), it would be difficult to reduce the program to less than 30 lines of code. The use of *PLC emulation techniques*, as demonstrated below, allows the same problem to be solved with a 5-line program!

When tackling the above mentioned problem with a PLC, the first step would be to graphically represent the system using a *ladder diagram*, as shown in **Fig. 1**.

A ladder diagram consists of one or more 'rungs' which are connected horizontally between the positive and negative supply rails. A symbol, similar to that of a capacitor, is used to indicate a pair of normally-open switch contacts, while the circle refers to an output device such as a lamp. The series arrangement of the two switches means that both sets of contacts must be closed for the lamp to light, which is a logical AND condition.

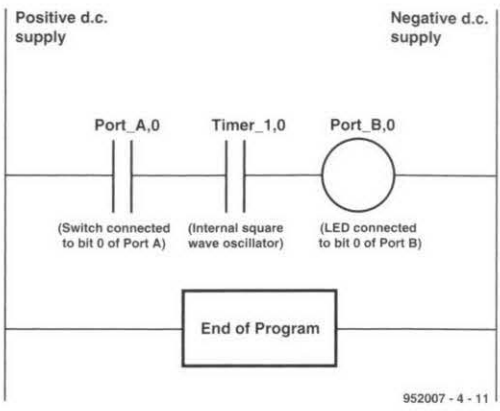


Fig. 1. LED flasher ladder diagram.

The first switch represents the state of input port bit RA0, with a logic 1 equivalent to the switch being pressed. The second switch is really a single bit of a counter/timer register, which is internal to the microcontroller. This particular bit turns 'on' and 'off' every 256 µs, thus producing a 2-kHz square wave signal. Finally, the LED output is assumed to be connected to bit RB0. **Listing 1** demonstrates how this function would be coded, based on the previous ladder diagram.

```
include "plc.h"          ;File containing PLC command definitions

ld       Port_A,0        ;Read input switch
and      Timer_1,0       ;Perform logical AND with 2 kHz squarewave
out      Port_B,0        ;Switch the LED on or off

endp                     ;End of the program
```

Listing 1. LED flasher program.

Commands such as *ld*, *and*, *out* and *endp* are stored as macro definitions in the assembler header file 'plc.h'. During assembly, the PIC's assembler (MPALC) expands each high level statement into several individual assembly language instructions. Thus the object code file produced by the assembler is much larger than the original source code file would suggest. The main saving is therefore in the time spent by the designer in solving the problem, rather than anything else!

A practical difficulty which might be encountered with the above program is that the external switch may generate a logic 0 when pressed, rather than a logic 1 which is assumed. Similarly, it might be necessary to output a logic 0 to cause the LED to light. These problems may easily be overcome by replacing the *ld* and *out* instruction with their negative logic equivalents *ld_not* and *out_not*, as shown in **Listing 2**.

A signal which is active low may be thought of as a switch whose contacts are

```
include "plc.h"          ;File containing PLC command definitions

ld_not  Port_A,0         ;Read input switch
and     Timer_1,0        ;Perform logical AND with 2 kHz squarewave
out_not Port_B,0         ;Switch the LED on or off

endp                     ;End of the program
```

Listing 2. LED flasher program with active low input and output signals.

normally closed. With this arrangement, pressing the switch causes the contacts to open, thus breaking the connection. In ladder diagrams, a pair of normally closed switch contacts is drawn as two vertical lines with a diagonal line across the switch symbol. Practical applications of normally closed switches will be seen shortly.

It may also be necessary to alter the pulse repetition rate of the flashing LED from the original 2 kHz. The PLC emulation software causes two 8-bit registers called *Timer_1* and *Timer_2* to be regularly incremented based on the microcontroller's clock signal. Assuming a 4-MHz clock frequency, periodic times in the range 512 µs to 16.77 seconds are available, as given in **Table 1**.

Timer_1		Timer_2	
Bit position	**Periodic time**	**Bit position**	**Periodic time**
0	512 µs	0	131.1 ms
1	1.024 ms	1	262.1 ms
2	2.048 ms	2	524.3 ms
3	4.096 ms	3	1.049 s
4	8.192 ms	4	2.079 s
5	16.38 ms	5	4.194 s
6	32.77 ms	6	8.389 s
7	65.54 ms	7	16.77 s

Table 1. Square wave signals available from Timer_1 and Timer_2.

Note: As will be seen later, a repeating pulse waveform may also be used as an input signal to a software *counter*, which will produce a logic 1 when the required number of pulses have been detected. The combination of a counter and a regular square wave signal allows time intervals of almost any size to be generated.

Another example may require that a lamp should be illuminated when one or more input sensors becomes active. The ladder diagram for this system is shown in **Fig. 2**.

The above arrangement may be thought of as allowing power to reach the lamp

if either or both of the input switches is closed, which is an inclusive-OR condition. The actual program for this arrangement is given in **Listing 3**.

So far, the logical operators AND and OR have been encountered. It is also possible to invert a signal using the *ld_not* instruction, as shown in **Fig. 3**.

Recall that the use of normally closed switch contacts here means that the output will be 'off' when the input switch is pressed, and 'on' when the switch is released. This is equivalent to a logical NOT operation. In fact, this effect can be achieved in two ways, — either by

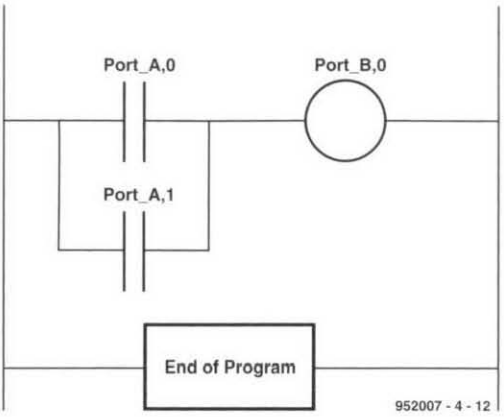


Fig. 2. Logical OR arrangement of input switches.

```
include "plc.h"          ;File containing PLC command definitions

ld      Port_A,0         ;Read input RA0
or      Port_A,1         ;Perform inclusive OR with bit RA1
out     Port_B,0         ;Output result to bit RB0

endp                     ;End of the program
```

Listing 3. Inclusive-OR arrangement of two inputs.

using the *ld_not* instruction followed by *out* — or by using *ld* followed by *out_not*. Where a logical operation such as AND and OR must be performed with a negated variable, this action may be performed by a single instruction, such as *and_not* or *or_not*. These instruction assume that the second variable is 'true' when at a logic 0 level.

The last type of logic function which is supported by the PLC software is the exclusive-OR operation, which is shown as a ladder diagram in **Fig. 4**.

Although the ladder diagram seems complex, the actual program (which uses the *xor* instruction) is

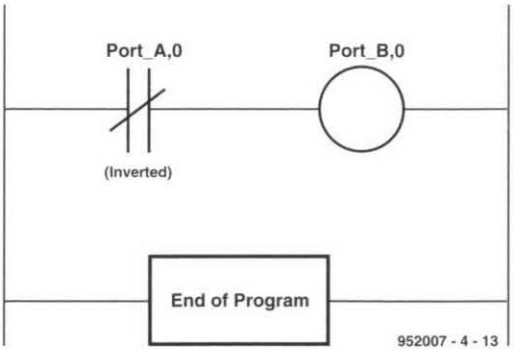


Fig. 3. Logical inversion using the ld_not instruction.

quite simple, as shown in **Listing 4**. As you may already have guessed, an *xor_not* is also available, which is an inverted form of the *xor* command.

When dealing with complex logic networks, such as the one shown in **Fig. 5**, it is sometimes necessary to calculate each branch separately, and then combine the results together to produce the actual output.

Some PLCs use 'block' logic functions such as *and_blk* or *or_blk* to enable complex logic networks to be solved. This approach is not used by the present PLC emulation software. Instead, two 8-bit registers called *Aux_1* and *Aux_2* are provided,

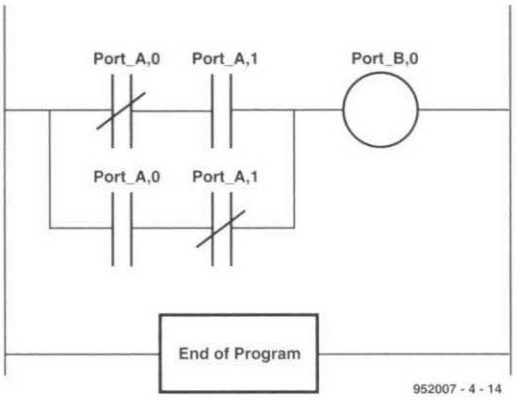


Fig. 4. Exclusive-OR function.

```
include "plc.h"          ;File containing PLC command definitions

ld      Port_A,0         ;Read input switch
xor     Port_A,1         ;Exclusive-OR with bit RA1
out     Port_B,0         ;Output result to bit RB0

endp                     ;End of the program
```

Listing 4. Exclusive-OR program.

which may be used to hold temporary results, such as the state of a particular branch of the ladder program. A total of 16 individual bits (sometimes referred to as *auxiliary relays* by PLC programmers) are therefore available for the storage and retrieval of partially calculated results. **Listing 5** demonstrates the use of a single auxiliary relay (Aux_1,0) as a temporary storage location when solving the previously seen two-branch logic function.

To summarize, **Table 2** gives a list of combinational logic instructions supported by the PLC emulation software instruction set, together with the commonly used ladder diagram symbol in each case.

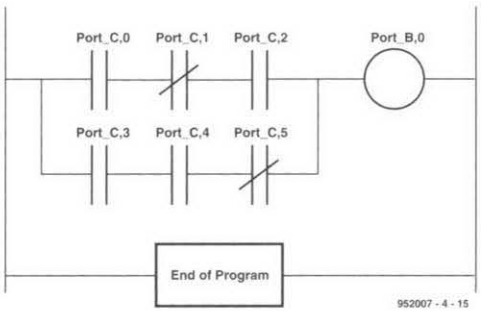


Fig. 5. Complex logic function requiring separate calculation of each branch.

```
include "plc.h"          ;File containing PLC command definitions

ld       Port_C,0        ;Read input RC0
and_not  Port_C,1        ;AND with (inverted) bit RC1
and      Port_C,2        ;AND with bit RC2
out      Aux_1,0         ;Store temporary result in bit Aux_1,0

ld       Port_C,3        ;Read input RC3
and      Port_C,4        ;AND with bit RC4
and_not  Port_C,5        ;AND with (inverted) bit RC5
or       Aux_1,0         ;OR the two branches together
out      Port_B,0        ;Send the result to bit RB0

endp                     ;End of the program
```

Listing 5. Using an auxiliary relay when solving a complex network.

Instruction syntax	Function performed	Ladder diagram symbol
ld register,bit	Read input	
ld_not register,bit	Read input (inverted)	
or register,bit	Inclusive-OR	
or_not register,bit	Inclusive-OR (inverted)	
and register,bit	Logical AND	
and_not register,bit	Logical AND (inverted)	
xor register,bit	Exclusive-OR	
xor_not register,bit	Exclusive-OR (inverted)	
out register,bit	Send to output	
out_not register,bit	Send to output (inverted)	

Table 2. Combinational logic instructions and ladder diagram symbols.

Latches, Counters and Timers

Simple latching circuits

Surprisingly, it is possible to perform logical operations which combine the current state of inputs and outputs. This results in feedback being applied, which is the basis of all sequential logic. As an example of this principle, the ladder diagram of **Fig. 6** shows how a simple set-reset bistable may be formed.

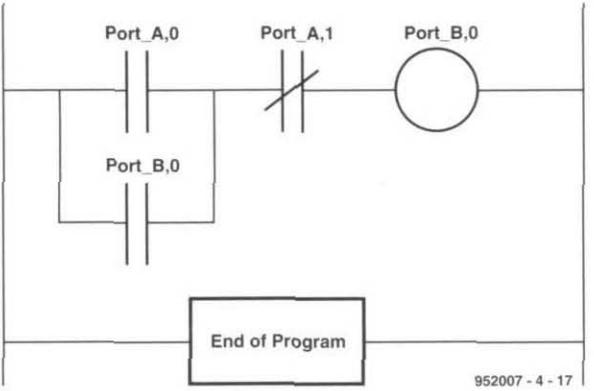


Fig. 6. Set-reset bistable, produced by applying feedback from outputs back to inputs.

The input signal RA0 is equivalent to the Set input of the bistable, while bit RA1 functions as a Reset pin. Notice that RA1 acts as a normally closed switch, so a connection exists between the input and the output under normal circumstances. RB0 acts as the Q output, which will follow RA0, when the bistable is initially set. Once the bistable has been set (Q=1), it will stay in this state, due to the ORed arrangement of RA0 and RB0. When input RA1 becomes logic 1, the self-latching arrangement is broken, and the bistable is reset. **Listing 6** shows how the latch function would be coded.

If required, a Q output may also be provided. This would be achieved by first reading the state of the Q output and then outputting this value in inverted form.

```
include "plc.h"          ;File containing PLC command definitions

ld       Port_A,0        ;Read input RA0 (Set input)
or       Port_B,0        ;Self latching arrangement
and_not  Port_A,1        ;Cancel latch if RA1 = 1 (Reset input)
out      Port_B,0        ;Output to RB0 (Q output)

endp                     ;End of the program
```

Listing 6. Using feedback to produce a set-reset bistable.

A typical command sequence might be *ld_not Port_B,0* followed by *out Port_B,1*.

The same basic principle may be used to construct a *clocked SR bistable*, whose outputs are updated only when the clock enable input is at the appropriate logic level. This is achieved by inserting a pair of AND gates in series with the Set and Reset inputs, as shown in **Listing 7**.

```
include "plc.h"          ;File containing PLC command definitions

ld       Port_A,0        ;Read input RA0 (Set input)
and      Port_A,2        ;AND with clock enable input
out      Aux_1,0         ;Store temporary result in Aux_1,0

ld       Port_A,1        ;Read input RA1 (Reset input)
and      Port_A,2        ;AND with clock enable input
out      Aux_1,1         ;Store temporary result in Aux_1,1

ld       Aux_1,0         ;Read (clocked) Set input
or       Port_B,0        ;Self latching arrangement
and_not  Aux_1,1         ;Cancel latch if Aux_1,1 = 1 (Clocked Reset)
out      Port_B,0        ;Output to RB0 (Q output)

endp                     ;End of the program
```

Listing 7. Clocked SR bistable.

Counters

The PLC emulation software provides eight individual counters, each of which can count up to a maximum value of 255. Each counter has two inputs and a single output, as shown in **Fig. 7**.

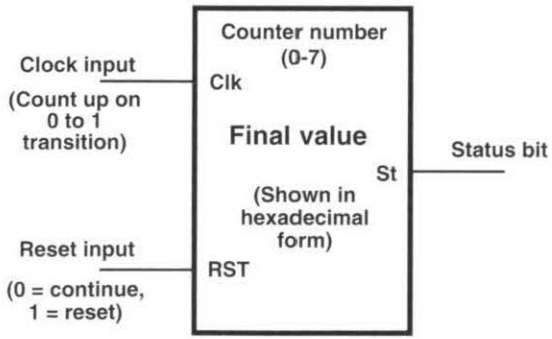


Fig. 7. Counter symbol shown in ladder diagram form.

Counters are very useful devices which may be used to produce time delays, single pulses, or to count external events. It is also possible to 'cascade' two or more counters to produce repeating pulse waveforms with various mark to space ratios.

Under normal circumstances, the Clock input to the counter is connected to a rung of the ladder diagram, which may be driven from an external input, a timer signal, or from some other source. Each 0-to-1 transition on this input causes an internal counter to be incremented (which is not directly visible to the program). The counter unit contains a preset 'final' or 'threshold' value, which is fixed at programming time. Once the internal count reaches this final value, the counter's status bit is set to one, and further input pulses have no effect. The counter may be *cleared by applying a logic 1 to the Reset input, which* clears the Status bit and also resets the internal counter back to zero. Once the counter has been reset, it may then be used to count pulses once again.

A single 8-bit register (*Ctr*) is used to hold the status bits of each counter. The least significant bit (Ctr,0) contains the current state of counter 0, through to the most significant bit (Ctr,7), which holds the status bit of counter 7. As mentioned above, a one in a particular status bit indicates that the required number of input pulses have ben detected, while a zero means that the count is still in progress. Notice from this description that the information provided by the counter is very basic — either the count is finished, or it hasn't. As we will soon see, this is all you need to know in the majority of counter/timer applications!

Example

The ladder diagram of **Fig. 8** shows a system which counts ten pulses on input RA0, before setting output bit RB0. The system may be reset by applying a logic 1 to input RA1.

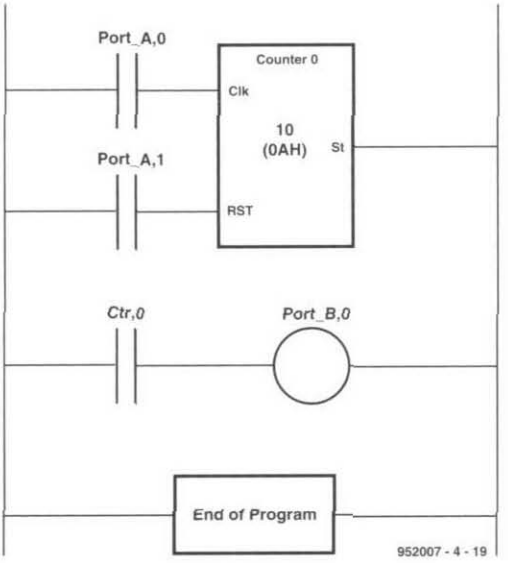


Fig. 8. Simple counter circuit.

Listing 8 shows how this would be written in program form.

There are several important points in the listing. Firstly, notice that the first rung is terminated by the *ctr* instruction, rather than by the more usual *out* or *out_not*. Thus the clock input to the counter is defined by the calculated result of the previous 'rung'. The ctr instruction itself has four arguments. These are the counter number (0 through 7), the final value (0 through 255), the reset register and finally the particular bit of the reset register (0 through 7) which is used to reset the counter. Finally, the current state of counter 0 is determined by reading the same bit of the *Ctr* register.

```
include "plc.h"                ;PLC command definitions

ld      Port_A,0               ;Read input RA0 (Clock input)
ctr     0,0AH,Port_A,1         ;Counter 0, final value = 10 (0AH)
                               ;Reset using input bit RA1

ld      Ctr,0                  ;Read counter 0 status bit
out     Port_B,0               ;and output to RB0

endp                           ;End of the program
```

Listing 8. Simple counter program.

Time delays

One problem which may be encountered with the above counter circuit, particularly if the clock input is produced by an input switch, is that of *contact bounce*. This mechanical effect may cause a single keypress to appear as a rapid burst of pulses, thus causing the counter status bit to become set prematurely. Although hardware debounce circuits are available, it is generally more economical to use a software time delay to perform this function. By driving the clock input of a counter from an appropriate timer register bit (register *Timer_1* or *Timer_2*), in combination with the use of a suitable final value for the counter, time delays of almost any size may be generated. For keyboard debounce applications, a time delay of around 20 ms is generally adequate, which could be produced by counting 20 pulses, each of 1 ms duration.

The method used here is commonly referred to as an *on-delay timer*. The circuit is arranged so that each time the key is pressed, the counter/timer is enabled. Releasing the key produces the opposite logic level, which is used to reset the timer. Brief pulses, such as those caused by contact bounce, cause the timer to be repeatedly enabled and cleared, which cannot lead to the status bit being set. Only by holding down the key for the required period of 20 ms will the timer produce a logic 1 output. The status bit of this timer can then be used as a denounced clock input waveform. **Listing 9** shows the required additions to add a keyboard debounce facility to the previously seen counter design.

```
include "plc.h"                    ;PLC command definitions

;Use counter 1 to provide a 20 ms time delay, based on RA0

ld      Timer_1,1                  ;Count 1 ms pulses when RA0 = 1
ctr     1,14H,Aux_1,0              ;Counter 1, final value = 20 (14H)
                                   ;Reset using input bit Aux_1,0

;Reset counter 1 if RA0 = 0

ld_not  Port_A,0                   ;Read input RA0 (inverted)
out     Aux_1,0                    ;Use Aux_1,0 to reset counter 1

;Use counter 1 status bit as a debounced input to counter 0

ld      Ctr,1                      ;Read debounced input from Ctr,1
ctr     0,0AH,Port_A,1             ;Counter 0, final value = 10 (0AH)
                                   ;Reset using input bit RA1

ld      Ctr,0                      ;Read counter 0 status bit
out     Port_B,0                   ;and output to RB0

endp                               ;End of the program
```

Listing 9. Counter program with denounced keyboard input.

Pulse generation

Another application of time delays (using counters) is the production of pulses of fixed duration. This is functionally similar to a monostable circuit.

The method used here is to use an externally generated trigger pulse to set a bistable. Thus, a brief pulse on the Set input causes the Q output of the bistable to go logic 1, and remain there once the pulse has ended. The Q output is used in turn to enable a timer, which should be configured to produce the required time interval. Once the required pulse width has been generated, the appropriate counter status bit becomes set, which may then be used to reset the latch. Thus the latch is set by the initial pulse, and is then cleared once a predetermined period of time has elapsed.

Before the monostable can be retriggered, it is necessary to reset the counter. This may be achieved manually with the external input, or automatically by connecting the $\overline{Q}$ output of the bistable to the counter's reset pin. The required ladder diagram for the monostable is shown in **Fig. 9**, while the actual program is given in **Listing 10**.

The ability to generate time delays and pulses is useful in many electronic designs. A security system would be a typical example, with the requirement for exit and entry delays to enable arming and disarming of the system. The external siren would also need to be disabled once a fixed time interval had elapsed, to avoid undue annoyance to neighbours or passers-by.

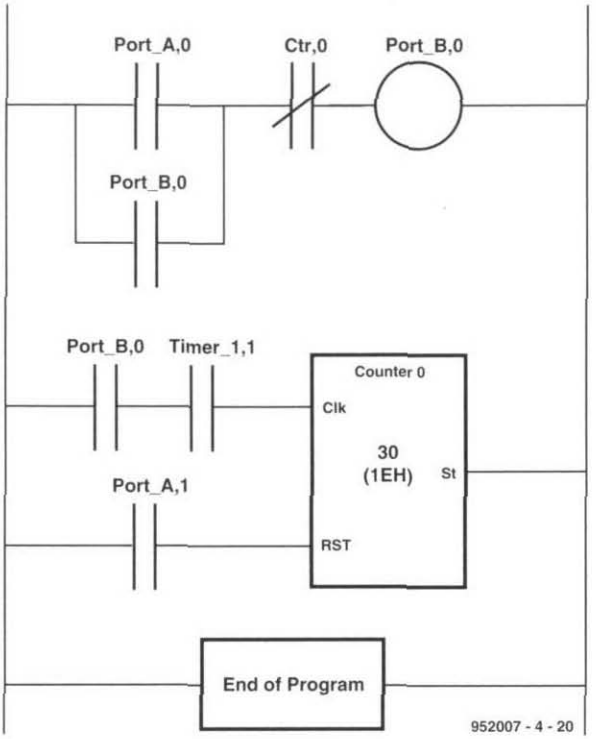


Fig. 9. Producing a single pulse using a latch and a counter.

```
include         "plc.h"          ;PLC command definitions

ld              Port_A,0         ;Use pulse on RA0 to set bistable
or              Port_B,0         ;Self latching arrangement
and_not         Ctr,0            ;Reset latch after time interval
out             Port_B,0         ;Output to RB0 (Q output)

ld              Port_B,0         ;Read Q output
and             Timer_1,1        ;Count if Q = 1 (512 us pulses)
ctr             0,1EH,Port_A,1   ;Counter 0, final value = 30 (1EH)
                                 ;Reset using input bit RA1

endp                             ;End of the program
```

Listing 10. Monostable program with manual counter reset.

Shift Registers and the Drum Sequencer

Shift registers

The PLC emulation software provides four shift registers, each of which is eight bits wide. The ladder diagram symbol and functional operation of a shift register are shown in **Fig. 10**.

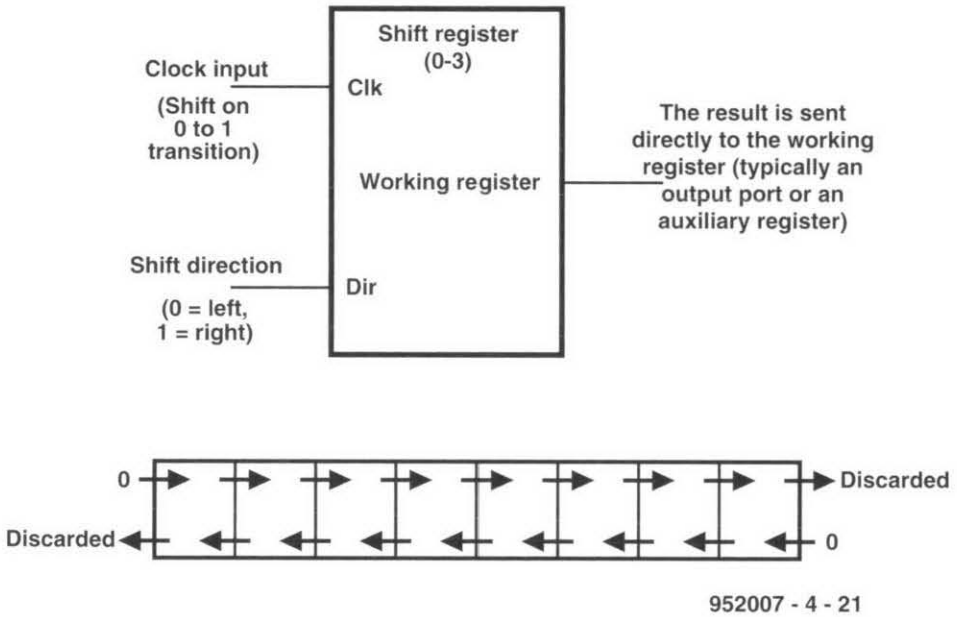


Fig. 10. Shift register symbol and functional operation.

As can be seen, the ladder diagram symbol for a shift register is similar to that of a software counter (considered earlier). The main differences are the replacement of the Reset input with a *direction select* control, and the absence of a Status output. Most of the programming techniques used with counters also apply to shift registers, including the use of a separate rung of the ladder diagram to drive the Clk input. Each 0 to 1 transition on this input causes the content of the working register to be shifted by one bit position to the left or right, as selected by the logic level present on the Dir input (0=shift left, 1=shift right).

The Dir input may be driven from an external input, or from a calculated function, thus allowing the shift direction to be altered during program operation. For situations where the required direction is known in advance, two special purpose

Register	Bit	Logic level	Possible applications
Temp_2	5	Logic 1	Shift register — shift right always Output port — permanent logic 1
Temp_2	4	Logic 0	Shift register — shift left always Output port — permanent logic 0 Counter — disable reset input

Note: the remaining bits of register *Temp_2* are reserved and must not be used

Table 3. Read-only auxiliary relays and their possible values.

(read-only) Auxiliary relays are provided, as given in **Table 3**.

In either operating mode, a 0 is shifted-in at one end, while another bit is being simultaneously discarded at the opposite end of the register. This behaviour was shown diagrammatically in Fig. 10.

Some mechanism is required to inject actual data into the shift register. The normal approach is to drive a single bit of the working register with the required logic level. When shifting to the right, the most significant bit of the working register would normally be used as an input bit, while the least significant bit would be used when shifting to the left. Each clock pulse then causes data to be shifted further into the working register. **Figure 11** shows the ladder diagram of a simple shift register application, while **Listing 11** shows how this would be coded.

Consider the detailed operation of this program. The first rung causes a square wave signal (*Timer_1,3*) to be applied to output pin RB7. This signal has a periodic time of approximately 4 ms, thus alternating between 1 and 0 every 2 ms. Shift register 0 is driven by *Timer_1,0* (which oscillates four times faster than *Timer_1,3*, and uses *Port_B* as its working register. The use of auxiliary register *Temp_2,5* as the Dir input causes the data to be shifted to the right on each 0 to 1 transition of *Timer_1,0*. Notice that both sections of the ladder diagram modify *Port_B*. The *out* instruction generates a square wave signal on the left-most bit (RB7), which is then shifted to the right every 0.5 ms. Thus a repeating pattern of four 'ones' followed by four 'zeros' is produced. By varying the periodic times of the two square wave signals, a range of moving display patterns may be generated.

Example

The following program demonstrates the use of a shift register to generate a more complex moving display pattern, which shifts a single bit to the left, and then to the right in a repeating sequence. This is illustrated by **Fig. 12**.

The solution presented here makes use of some combinational logic, two counters, a bistable and a shift register. Although this may seem complex, each section has a well defined function and can be considered in isolation.

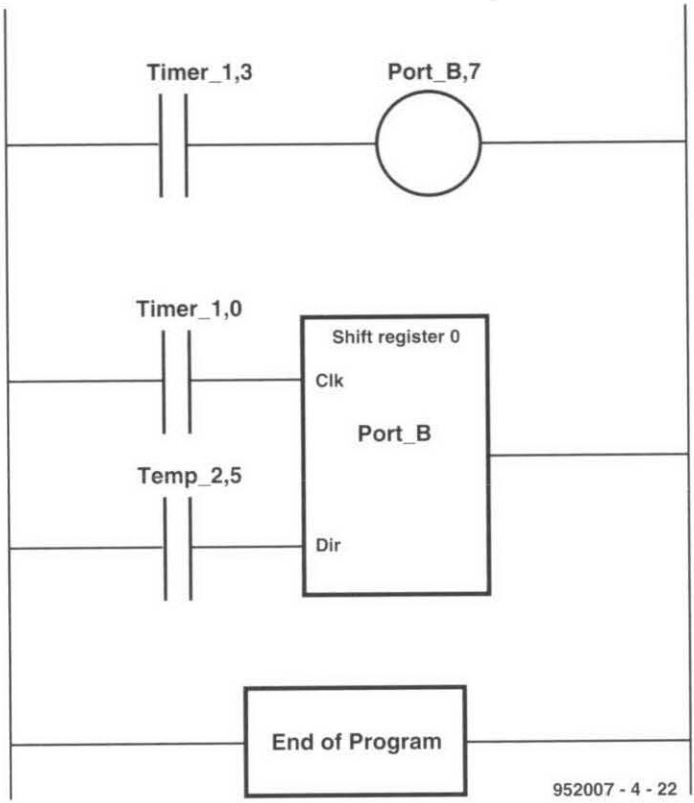


Fig. 11. Simple shift register ladder diagram.

```
include         "plc.h"              ;PLC command definitions

ld              Timer_1,3            ;4 ms waveform
out             Port_B,7             ;Output to MSB of Port_B (RB7)

ld              Timer_1,0            ;Shift register clock (0.5 ms)
shift           0,Temp_2,5,Port_B    ;Shift register 0
                                     ;Direction = right (Temp_2,5 = 1)
                                     ;Working (output) register = Port_B

endp                                 ;End of the program
```

Listing 11. Simple shift register program.

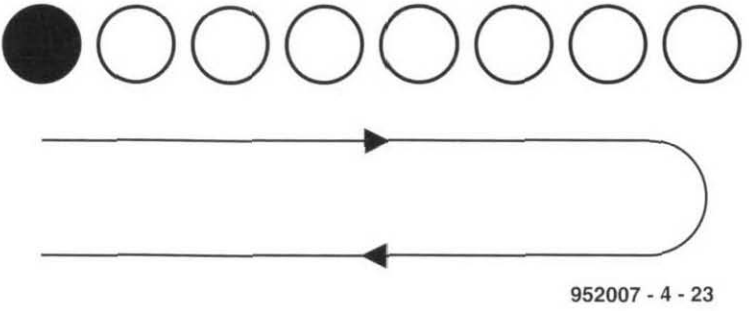


Fig. 12. The required running light display pattern.

- An exclusive-OR gate generates a logic 1 in the left-most bit of the output register, prior to shifting to the right. A second exclusive-OR gate performs the same function in the right-most bit, prior to shifting the register to the left.
- To clear the output register, it is necessary to shift the register to the right (or to the left) at least eight times. Counter 0 is therefore used to count the number of steps to the right, while counter 1 counts steps to the left.
- A set-reset bistable is arranged so that the latch is set by the counter 0 status bit (Ctr,0), and then cleared by the counter 1 status bit (Ctr,1). The reset inputs of each counter are connected to the bistable outputs in such a way that counter 1 is reset while counter 0 is active, and vice versa. Thus the two counters run alternately, in a repeating sequence.
- The Dir input to the shift register is driven from the Q output of the bistable. This causes the display pattern to be shifted to the right when counter 0 is active, and to the left when counter 1 is active.

Figure 13 shows typical logic levels generated by each section, while **Listing 12** gives the actual program.

The Drum Sequencer

Drum sequencers are particularly useful in sequential control applications, where several output devices must be activated in a pre-defined sequence. A typical application would be an automated production line where the same series of operations must be performed on each component. The functional equivalent of a simple drum sequencer is shown in **Fig. 14**.

This particular drum has four individual output bits and can be rotated in 90° steps, giving a maximum sequence length of four steps (after which the drum will have returned to its start position). The drum may also be returned to the start position at any time by applying a pulse to the Reset input. When represented in ladder diagram form, a symbol is used which closely resembles that of a counter — look ahead to Fig. 15 for an example.

The PLC emulation software provides an 8-bit wide drum sequencer, which may have up to eight steps. Eight contiguous memory locations (registers) are set aside to hold the content of the drum, and these must be initialized with the required

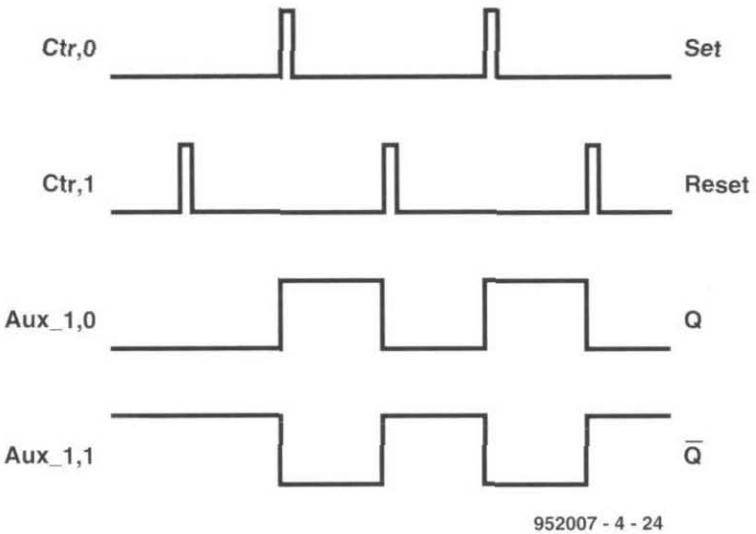


Fig. 13. Internal logic levels produced by the running light display program.

data before the drum is used. This initialization is performed by the *preset* instruction, which accepts two arguments. The first argument is the name of the register, which corresponds to one of the eight drum positions, and the second is the actual drum content. Register *Dr_data* is the first location. Other locations are accessed by adding an offset to this base address, so the last data location, for example, would be referred to as register *Dr_data+7*. At any time, the output of the drum may be obtained by reading the content of register *Drum*. Thus *Drum,7* represents the most significant bit of the drum, while *Drum,0* is the least significant bit.

As was the case with counters and shift registers, the Step (or Clock) input to the drum is provided by the previous rung of the ladder diagram. Thus the drum instruction appears at the end of the rung, in place of the more usual *out* or *out_not* instruction. The drum instruction itself accepts two parameters, which are the name of the Reset register, followed by the Reset bit (0-7). **Figure 15** shows the ladder diagram of a simple drum sequencer application, which produces a moving display pattern on output port bits RB0-RB7. The actual program is given in **Listing 13**.

Listing 13 is quite straightforward, and may be considered in three separate parts. The first section initializes the eight locations used by the drum, with the required display pattern. Register *Dr_data* holds the first line, which is assumed to corresponds to 'all LEDs off'. This is also the line which will be displayed when the drum

```
include         "plc.h"             ;PLC command definitions

;Use counter 0 to count 8 steps to the right

ld              Timer_1,0           ;0.5 ms waveform
ctr             0,8,Aux_1,0         ;Counter 0, final value = 8
                                    ;Reset using Q output of latch

;Use counter 1 to count 8 steps to the left

ld              Timer_1,0           ;0.5 ms waveform
ctr             1,8,Aux_1,1         ;Counter 1, final value = 8
                                    ;Reset using Not Q output of latch

;Use latch to switch shift register direction left and right every 8
 steps

ld              Ctr,0               ;Counter 0 status Sets the latch
or              Aux_1,0             ;Self latching arrangement
and_not         Ctr,1               ;Counter 1 status Resets the latch
out             Aux_1,0             ;Q output

ld_not          Aux_1,0             ;Read Q output (inverted)
out             Aux_1,1             ;Not Q output

;Generate a logic 1 in left/right bit following direction change
;(This bit is then 'shifted' across the output port)

ld              Port_B,0            ;Invert RB0 when counter 0 pulse
xor             Ctr,0               ;is detected
out             Port_B,0            ;Output to RB0

ld              Port_B,7            ;Invert RB7 when counter 1 pulse
xor             Ctr,1               ;is detected
out             Port_B,7            ;Output to RB7

;Left/right shift register connected to Port_B

ld              Timer_1,0           ;Shift register clock (0.5 ms)
shift           0,Aux_1,1,Port_B    ;Shift register 0
                                    ;Direction from Aux_1,1 (Not Q)
                                    ;Working (output) register = Port_B

endp                                ;End of the program
```

Listing 12. Running light display program.

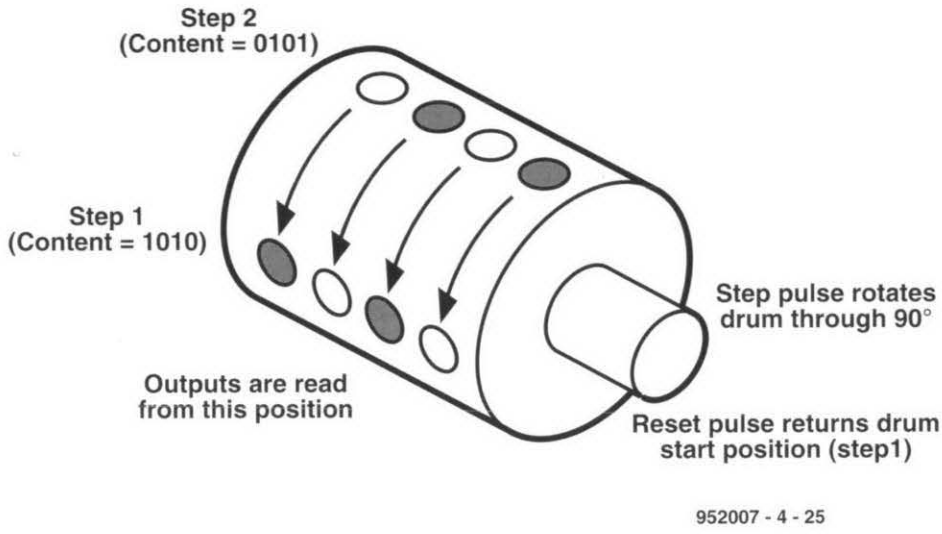


Fig. 14. 4 x 4 drum sequencer — simplified operation.

is reset by applying a logic 1 to input RA0. Following lines specify the actual display pattern. The second section of the program controls the drum sequencer itself, so in this case, the drum rotates every 512 µs, and may be reset using input bit RA0. Lastly, the eight output bits of the drum are copied individually from register *Drum* to *Port_B*.

Example

The next program demonstrates a more complex use of the drum sequencer — bidirectional stepper motor control. In this application it is assumed that output bits RB0-RB3 are connected, via a stepper motor driver IC, to the coils of a four-phase unipolar stepper motor. The stepper motor is then controlled using two input bits which provide direction select (RA0) and single-step (RA1) functions. In other words, each 0 to 1 transition on the Step input causes the stepper motor to rotate through a certain angle in the direction selected by the Dir input. The program's function is to generate an appropriate sequence of output codes to drive the stepper motor coils, under the control of the Step and Dir inputs.

The approach used here is to store the output sequences for forward and reverse rotation in the upper and lower nibbles of the drum sequencer respectively. A combinational logic arrangement is then used to copy the correct set of output codes (forward or reverse) from the *Drum* register to output port bits RB0-RB3, based on the logic level present on the Dir input. Each pulse applied to the Step input then causes the drum to 'rotate', thus stepping the motor in the chosen direction. The actual program is given in **Listing 14**.

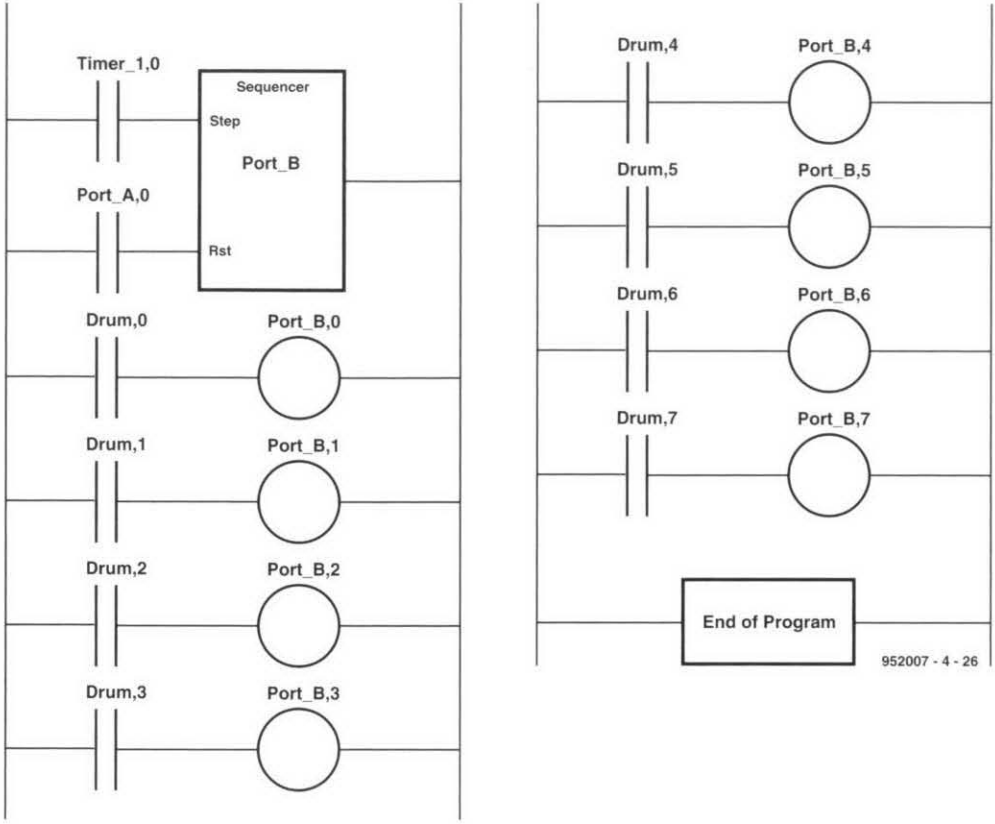


Fig. 15. Moving light display which makes use of the drum sequencer.

Clearly, the program in Listing 14 could form the basis of a more complex stepper motor controller. For example, by producing Step pulses in conjunction with a software counter, the position of the stepper motor may be controlled. Similarly, by varying the rate of Step pulses under program control, the motor could be accelerated or decelerated. Finally, as an alternative to the drum sequencer technique presented here, please bear in mind that the same effect could be achieved using a simple shift register program.

```
include         "plc.h"                 ;PLC command definitions

;Set up the content of the Drum Sequencer (8 rows x 8 bits)

preset          Dr_data,b'00000000'     ;Line 1 of the drum sequencer
preset          Dr_data+1,b'00011000'   ;Line 2 of the drum sequencer
preset          Dr_data+2,b'00100100'   ;Line 3 of the drum sequencer
preset          Dr_data+3,b'01000010'   ;Line 4 of the drum sequencer
preset          Dr_data+4,b'10000001'   ;Line 5 of the drum sequencer
preset          Dr_data+5,b'00000000'   ;Line 6 of the drum sequencer
preset          Dr_data+6,b'00000000'   ;Line 7 of the drum sequencer
preset          Dr_data+7,b'00000000'   ;Line 8 of the drum sequencer

;Rotate drum every 0.5 ms

ld              Timer_1,0               ;0.5 ms clock waveform
drum            Port_A,0                ;Drum sequencer
                                        ;Reset bit = Port_A,0

;Copy drum output bits to Port_B

ld              Drum,0
out             Port_B,0

ld              Drum,1
out             Port_B,1

ld              Drum,2
out             Port_B,2

ld              Drum,3
out             Port_B,3

ld              Drum,4
out             Port_B,4

ld              Drum,5
out             Port_B,5

ld              Drum,6
out             Port_B,6

ld              Drum,7
out             Port_B,7

endp                                    ;End of the program
```

Listing 13. Moving display pattern using the drum sequencer.

```
include         "plc.h"                 ;PLC command definitions

;Set up the content of the Drum Sequencer (8 rows x 8 bits)
;left nibble = rotate pattern right, right nibble = rotate pattern left

preset          Dr_data,b'01100110'     ;Line 1 of the drum sequencer
preset          Dr_data+1,b'00111100'   ;Line 2 of the drum sequencer
preset          Dr_data+2,b'10011001'   ;Line 3 of the drum sequencer
preset          Dr_data+3,b'11000011'   ;Line 4 of the drum sequencer
preset          Dr_data+4,b'01100110'   ;Line 5 of the drum sequencer
preset          Dr_data+5,b'00111100'   ;Line 6 of the drum sequencer
preset          Dr_data+6,b'10011001'   ;Line 7 of the drum sequencer
preset          Dr_data+7,b'11000011'   ;Line 8 of the drum sequencer

;Rotate drum after 0-1 transition on Step input (RA1)

ld              Port_A,1                ;Read Step input (RA1)
drum            Temp_2,4                ;Drum sequencer
                                        ;Reset bit (Temp_2,4)=0 always
;Copy appropriate drum output bits to RB0-RB3, based on Dir input (RA0)
;(RA0=0 - shift right, RA0=1 - shift left)

;Stepper motor coil 0

ld              Drum,0                  ;Rotate left (bit 0)
and             Port_A,0                ;AND with Dir
out             Aux_1,0                 ;Temporary store
ld              Drum,4                  ;Rotate right (bit 0)
and_not         Port_A,0                ;AND with Dir (inverted)
or              Aux_1,0                 ;OR branches together
out             Port_B,0                ;Drive stepper motor coil 0

;Stepper motor coil 1

ld              Drum,1                  ;Rotate left (bit 1)
and             Port_A,0                ;AND with Dir
out             Aux_1,1                 ;Temporary store
ld              Drum,5                  ;Rotate right (bit 1)
and_not         Port_A,0                ;AND with Dir (inverted)
or              Aux_1,1                 ;OR branches together
out             Port_B,1                ;Drive stepper motor coil 1

;Stepper motor coil 2

ld              Drum,2                  ;Rotate left (bit 2)
and             Port_A,0                ;AND with Dir
out             Aux_1,2                 ;Temporary store

ld              Drum,6                  ;Rotate right (bit 2)
and_not         Port_A,0                ;AND with Dir (inverted)
or              Aux_1,2                 ;OR branches together
out             Port_B,2                ;Drive stepper motor coil 2
```

```
;Stepper motor coil 3

ld              Drum,3              ;Rotate left (bit 3)
and             Port_A,0            ;AND with Dir
out             Aux_1,3             ;Temporary store

ld              Drum,7              ;Rotate right (bit 3)
and_not         Port_A,0            ;AND with Dir (inverted)
or              Aux_1,3             ;OR branches together
out             Port_B,3            ;Drive stepper motor coil 3

endp                                ;End of the program
```

Listing 14. Bi-directional control of a stepper motor using the drum sequencer.

3. Method of Operation

An understanding of the internal operation of the PLC emulation software can be useful when deciding whether or not to use a 'ladder logic' approach in a particular situation. Consider the ladder diagram of Fig. 1 and the associated listing 1 as a typical example.

As mentioned previously, each PLC command is actually defined as a macro definition in the assembler header file 'plc.h' Assembling the program causes each macro to be replaced by a short section of assembly language.

The first instruction *ld Port_A,0* reads bit RA0 and places the result into the least significant bit of the W register. The second instruction *and Timer_1,0* is performed in two distinct phases. First, the state of bit 0 of the *Timer_1* register is loaded into a temporary storage location. Then the two registers are bitwise ANDed, leaving the result once again in the W register. At the end of this 'rung', the instruction *out Port_B,0* causes the least significant bit of the W register to be copied to bit RB0. Finally, the statement *endp* marks the end of the program.

The *endp* macro actually cause the microcontroller to jump back the start of the program, which forces the program to run in a continuous loop. The operation of the software therefore consists of a repeating sequence of reading inputs and calculating outputs. As the complexity of the software grows, the time for a single *scan cycle* increases. For the applications considered here, the scan time is less than 0.5 ms. One consequence of this is that an input signal must be present for at least one scan cycle to guarantee that it will be 'seen' by the software. With complex ladder logic arrangements, it may take several scan cycles before a change in an input variable propagates fully through the system. This may place further restrictions on minimum input pulse widths, and may also lead to the production of transient 'glitches' during calculations. Care must be taken to ensure that these short-lived signals do not cause false triggering of external logic. In practice the restrictions are not a problem in the vast majority of cases, provided that the speed of operation of the software is significantly faster than the response time required

for the application.

The header file also performs other initialization tasks, such as defining the direction of the ports. By default, ports A and C (with the 16C55) are defined as inputs while port B is set as an output. This may be altered if required, simply by editing the appropriate section of the header file. The RTCC register is initialized to increment, based on the clock output frequency, using a prescaler value of 256. The RTCC register (*Timer_1*) automatically updates a second register (*Timer_2*) on overflow, effectively creating a 16-bit timer. The *watchdog timer* is also enabled, with an effective timeout of 18 ms. During normal program operation, the watchdog timer is regularly cleared, but in the event of a program 'crash', the microcontroller will be reset automatically.

4. Testing your designs

Using the Assembler and Simulator

A diskette, containing all program listings is available through the *Elektor Electronics* Readers Services, in combination with the Experimentation Board for PICs. Before using the software, it is suggested that you first copy all files into a subdirectory on the hard disk. A suitable (DOS) command sequence might be:

C:	- Select the hard disk as the default drive,
**CD **	- and move to the root directory.
MD PLC	- Create a new directory,
CD PLC	- and make this directory the default.
COPY A*.*	- Copy all files from the A: drive.

As an example, the following description illustrates the assembly and simulation of listing 1.

MPALC LIST_1	- Assemble 'LIST_1.ASM'.
MPSIM	- Enter the PIC simulator.
LO LIST_1	- Load the object code file 'LIST_1.OBJ'.
ST LIST_1	- Load the stimulus file 'LIST_1.STI'.
E	- Execute the program.
SS	- Single step.
RE	- Reset the elapsed timer to zero.
H	- Read the on-line help information.
Q	- Quit to DOS.

A similar procedure to that shown above may be used to load and simulate most of the programs covered in this supplement. Bear in mind, though, that the operating speed of the simulator is several orders of magnitude slower than a real system!

Creating your own programs

If you follow a few simple rules, the process of writing and assembling your own programs will be relatively free of errors. Here are a few suggestions.

- The first line of the program (excluding statements) should be *include "plc.h"*.
- Command names are entered entirely in lower case (*ld*, *and*, *out*, etc.).
- Register names should be typed exactly as given. In general, the first letter is capitalized (*Ctr,0*, for example).
- The first command in a rung is always either *ld* or *ld_not*.
- The last command in a rung is generally *out* or *out_not*. Exceptions are rungs which are connected to counters, shift registers or to the drum sequencer.
- The last instruction in the program must be *endp*.

Hardware design

Figure 16 shows a schematic diagram of a simple microcomputer, which may be used to test prototype 16C54 software designs, while **Fig. 17** gives the layout of the 'Experimentation board for PICs' which was originally described in the

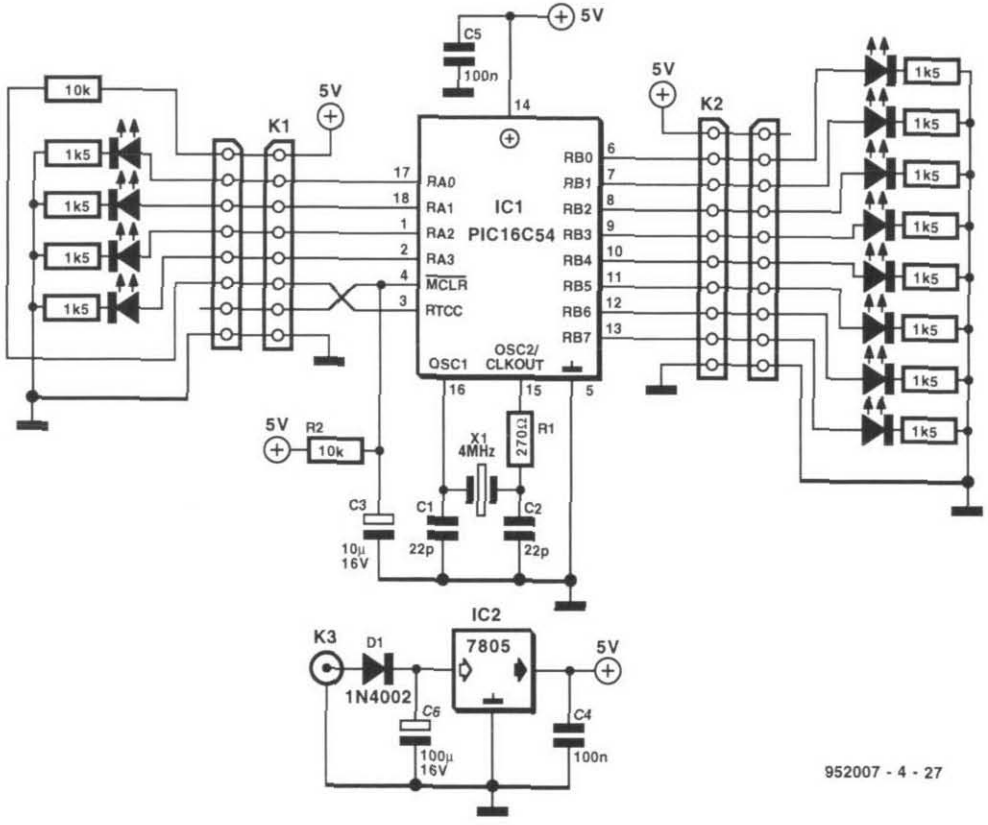


Fig. 16. Simple 16C54 based microcomputer with LEDs on all I/O ports. Note that the crystal frequency has been reduced from 12 MHz to 4 MHz.

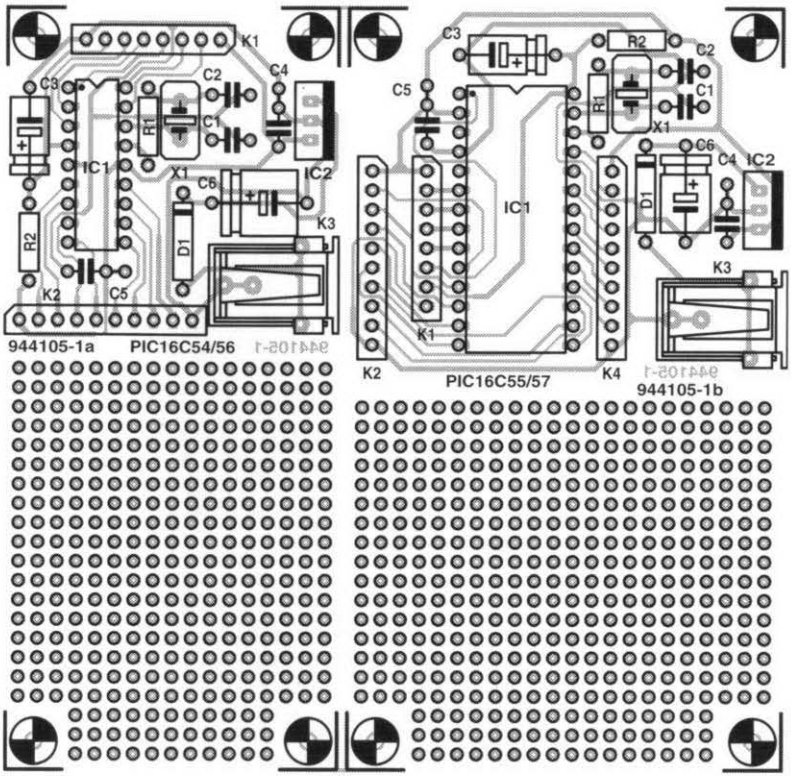


Fig. 17. Component overlay for the Experimentation Board for PICs (board available ready-made, see information at the end of section 'Summary').

July/August 1994 issue of *Elektor Electronics*, and which lends itself excellently to the construction of (experimental) PIC based hardware. The resistor arrays and the LEDs are mounted in the prototyping area of the board, and connected to the appropriate PIC pins via the SIL connectors and short wires.

The microcontroller itself requires a stabilized 5-V supply, which is provided by a 7805 regulator (IC2). It is assumed that the board will be powered from a low voltage (9-15 V and 1-100 mA) d.c. supply which may be produced by a suitable mains adaptor. Polarity reversal protection is afforded by diode D1.

To ensure correct operation of time critical software (particularly programs that use registers Timer_1 or Timer_2), a 4-MHz quartz oscillator circuit is used. Users should therefore select the 'XT' oscillator option when programming EPROM based PICs.

Components R2 and C3 provide a power-on reset facility, thus ensuring correct operation of the microcontroller following the application of power. Capacitor C5 acts as a decoupling device.

Each input or output port bit is connected to an LED status indicator, thus simplifying the process of software testing. The current limiting resistor value chosen is based on the assumption that low current LEDs (I_F=2 mA) will be used. Thus, the worst case current drain for Port B, due to the LEDs alone will be 16 mA (out of the 40 mA at logic 1 for the entire port).

Finally, the SIL header pins for external connection to ports A and B are intended to allow the PCB to be linked via a ribbon cable to special purpose input/output circuitry including switch units, relays, opto triacs, opto-isolators and Darlington transistors.

A Sample Application — Car Alarm

To illustrate the design process which is normally followed, a simple car alarm program will now be developed. The design of associated hardware is left as an exercise for the reader).

The first step is to decide on the required input and output signals, and to allocate these to particular microcontroller pins, as shown in **Fig. 18**.

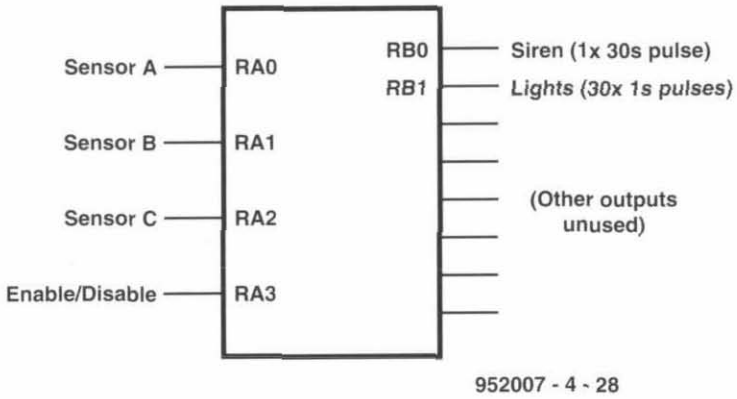


Fig. 18. A simple car alarm system.

Signal	Active level	Max. Current	To/From
Enable/disable	High = ON	n/a	From switch or IR receiver
Sensors A, B, C	High = ON	n/a	From opto-isolators
Siren	High = ON	1 mA	to 10-A relay via transistor
Lights	High = ON	1 mA	to 10-A relay via transistor

Table 4. Input/output schedule for the car alarm application.

Next, a table (or *input/output schedule*) is drawn up which specifies the electrical behaviour of each signal. A typical input/output schedule is given in **Table 4**.

Figure 19 shows a ladder diagram for the car alarm, while **Listing 15** gives the actual program.

The operation of the program is quite simple, and can be considered in four separate parts.

1. The three (active high) sensor inputs are inclusive ORed to give the current sensor status on bit Aux_1,0, provided the alarm is enabled.
2. The sensor status (Aux_1,0) is used to set a software latch. The Q output of this latch is used to drive the external siren, via a 10-A relay.
3. A software timer is used to generate a 30-s time delay, which is used to reset the latch after the bell timeout.
4. The lights are pulsed at 1-second intervals by ANDing the siren output with a 1-Hz square wave.

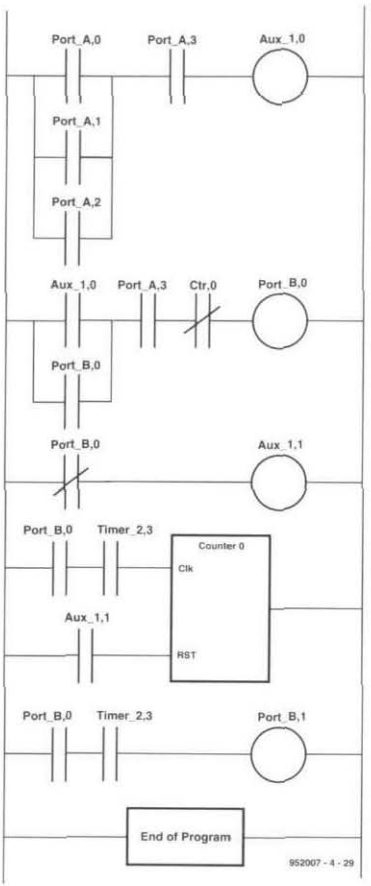


Fig. 19. Ladder diagram for the car alarm application.

5. Summary

Hopefully, my contribution to this Special Autumn 1995 Supplement helps to convince you that 'ladder logic' techniques can greatly simplify the design of control-oriented microcontroller-based systems! Although the PLC emulation software was developed primarily for use with the 16C54 and 16C55 microcontrollers, the same ideas could be applied to any microcontroller which offers similar hardware facilities to the 16Cxx family. The use of ladder logic allows the program's operation to be defined in terms of the required electrical behaviour of the system, so ladder logic is a programming language which is very natural to most electrical engineers. The fact that most instructions relate to the required behaviour of the system, rather than to the internal operation of the microcontroller, confirms that this is indeed a high level language.

```
;Note      Timer_1,3 is used here to speed up the simulation process.
;          Before programming an actual device replace BOTH references
;          with Timer_2,3.

include "plc.h"                ;PLC command definitions

;Determine the state of sensors A, B and C (if enabled)

ld         Port_A,0            ;Read sensor A
or         Port_A,1            ;Or with sensor B
or         Port_A,2            ;OR with sensor C
and        Port_A,3            ;AND with enable/disable bit
out        Aux_1,0             ;Output to Sensor Status bit

;Alarm activated latch

ld         Aux_1,0             ;Read sensor status bit
or         Port_B,0            ;Self latching arrangement
and        Port_A,3            ;Break latch if alarm disabled
and_not    Ctr,0               ;Reset latch after timeout(30s)
out        Port_B,0            ;Output to Siren bit (Q)
ld_not     Port_B,0            ;Read Siren bit (inverted)
out        Aux_1,1             ;Generate Not Q (to reset ctr0)

;30 second timer

ld         Port_B,0            ;Read Siren bit
and        Timer_1,3           ;and with 1 s pulse train
ctr        0,1EH,Aux_1,1       ;Counter 0, final value=30(1EH)
                               ;Reset using Aux_1,1

;Produce pulsed output for Light flasher circuit

ld         Port_B,0            ;Read Siren bit
and        Timer_1,3           ;AND with 1 s pulse train
out        Port_B,1            ;Output to Lights bit

endp                           ;End of the program
```

Listing 15. The car alarm program.

The diskette for 'PLC emulation using PIC microcontrollers' is available through the Readers Services. It is supplied free of charge with every order for the 'Experimentation board for PICs', order code 944105-1 (July/August 1994). For price and ordering information, please refer to the Readers Services page in your most recent issue of *Elektor Electronics*. This offer is subject to stock availability, and available until February 1996.

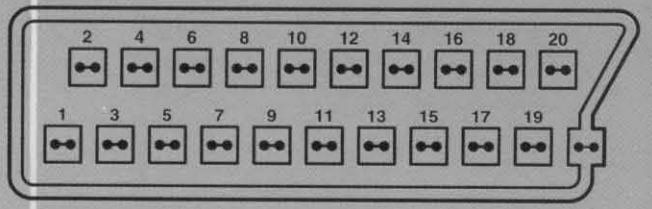
950103 - I

ELEKTOR ELECTRONICS

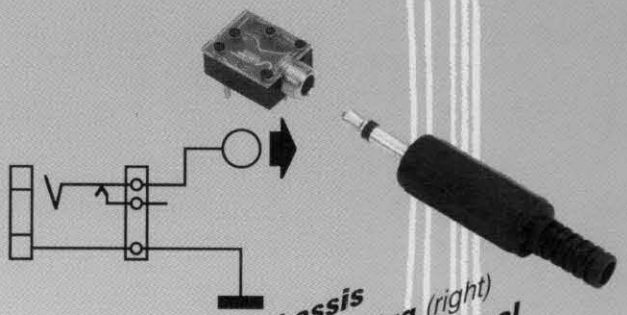
The International Electronics Magazine

THE AUDIO & VIDEO CONNECTION

PIN DESIGNATIONS FOR SCART CONNECTORS

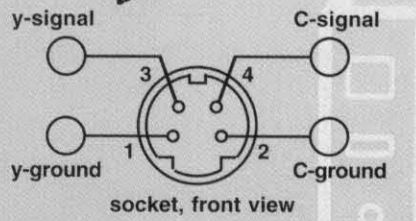


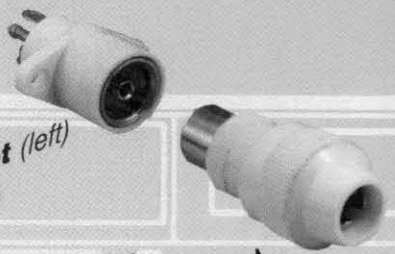
Pin		Pin	
1	R audio out	12	Data bus
2	R audio in	13	Chassis (R video)
3	L audio out	14	Chassis (data bus)
4	Chassis (audio)	15	R video in
5	Chassis (B video)	16	Fast video blanking
6	L video in	17	Chassis (composite video)
7	B video in	18	Chassis (fast video blanking)
8	Source switching	19	Composite video out
9	Chassis (G video)	20	Composite video in
10	Data bus	21	Chassis (casing)
11	G video in		



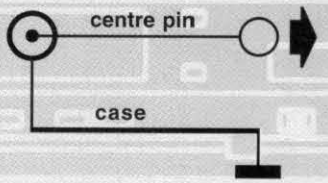
2.5 mm or 3.5 mm chassis jack socket *(left)* **and jack plug** *(right)*
Control connection with switch signal for camcorder *(in/out)*

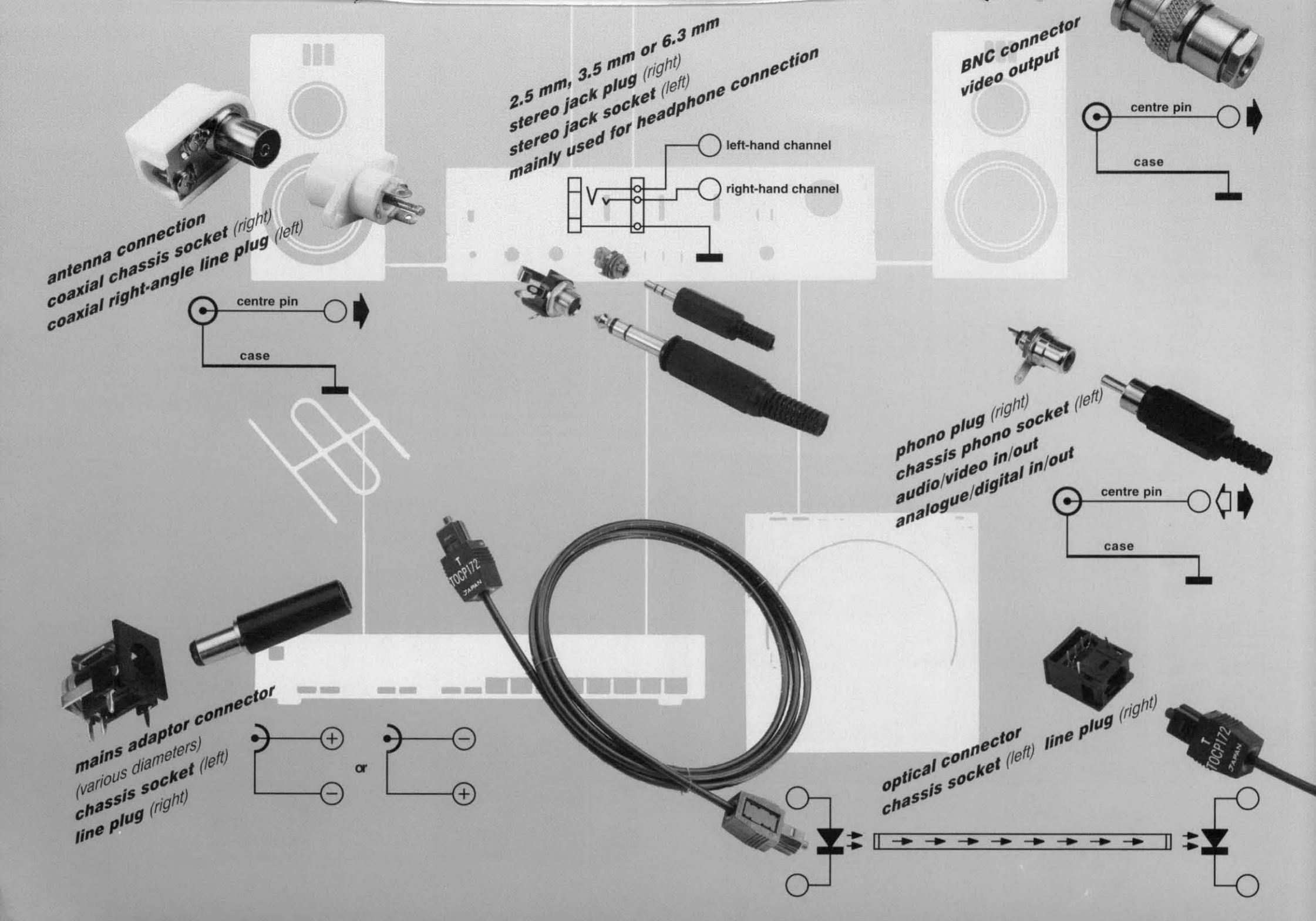
4-pin mini DIN plug s-video plug with isolated luminance (y) and chrominance (C) signals





antenna connection
coaxial chassis socket *(left)*
coaxial line plug *(right)*



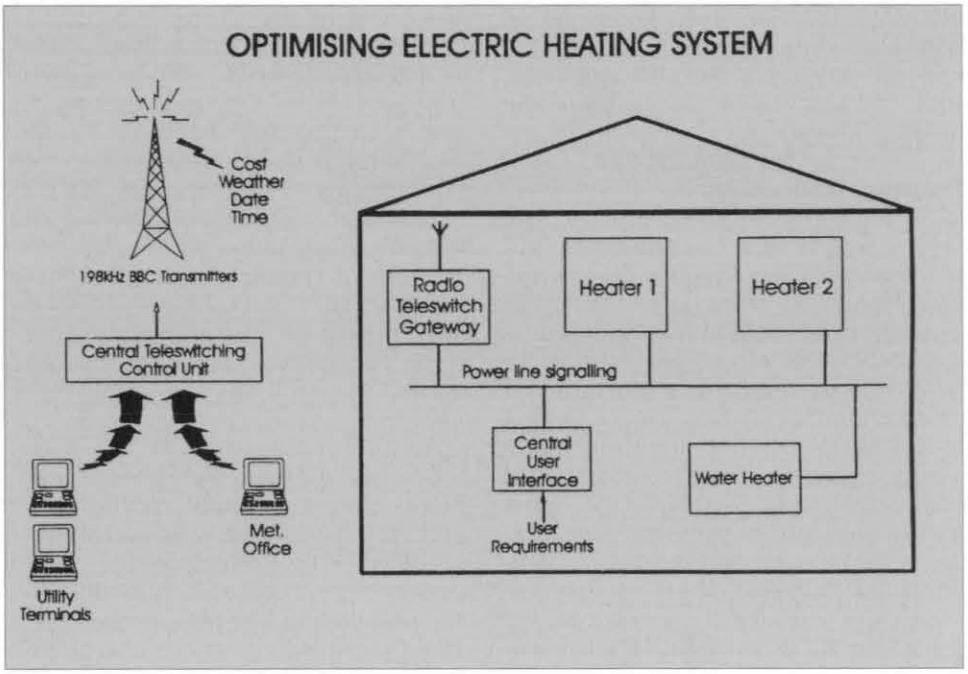

antenna connection
coaxial chassis socket (right)
coaxial right-angle line plug (left)
centre pin
case
2.5 mm, 3.5 mm or 6.3 mm
stereo jack plug (right)
stereo jack socket (left)
mainly used for headphone connection
left-hand channel
right-hand channel
BNC connector
video output
centre pin
case
phono plug (right)
chassis phono socket (left)
audio/video in/out
analogue/digital in/out
centre pin
case
mains adaptor connector
(various diameters)
chassis socket (left)
line plug (right)
or
optical connector
chassis socket (left) line plug (right)

From the World of Electronics

DIGITAL TELEVISION BY 2000?

Eagerly expected for some time, the Government has finally made public its proposals for new television channels using digital technology. Under the proposals, 20 new channels will be made available that will, however, occupy not much more of the frequency spectrum than the existing five channels. The present operators of the terrestrial analogue channels are automatically assured of new channels, but they will also be able to compete, with other companies, for one or more of the remaining fifteen channels.

There is, however, a drawback in all this for the viewer, because he or she will have to buy either an additional unit to receive the new services, which may, at present estimates, cost up to £ 500, or buy a new TV set, which, again at current estimates, may cost up to £ 2000. If the current quality of programmes is maintained, not many viewers may wish to avail themselves of the new services at those prices.



The EA Technology Celect System.

BT AND DIXONS EXPAND HIGH STREET PRESENCE

BT and Dixons have agreed to increase the number of 'The Link' high street communications shops from the eight opened in a trial which started in October 1994.

The first of the new stores – trading under the name 'Dixons – The Link' has already opened in Camden and shops in Croydon, Staines and Harrow will follow shortly. In all, about 50 new stores are planned for the current financial year, with more to open later.

The plans will mean further expansion of BT's high street presence and provide BT with a further opportunity to convince customers that their branded products and services have the leading edge in communications technology. Some 40 BT products will be sold at 'The Link', including fixed and mobile phones, fax machines, paging services and satellite systems. Customers will also be able to order new BT lines, chargecards, select service, and pricing options.

TOWARDS THE INTELLIGENT HOME

Although slow to gather momentum, the concept of the 'intelligent home' is becoming increasingly more substantive. The technology is available, and electronic, electric and construction companies, mainly through such organizations as the British Automated Homes and Buildings Association (BAHBA)[1], are cooperating to turn yesterday's philosophy into today's reality.

Such schemes are known as Home and Building Electronic Systems (HBES) and, indeed, there are many micro-computers and micro-controllers already in the modern British home. Almost every electrical appliance available today is controlled by some form of silicon chip, providing a certain degree of intelligence.

In the intelligent home, these devices are made interactive to maximize comfort and system efficiency. For instance, today's heating systems probably contain a boiler, a timeclock, thermostats and control valves. With an intelligent system, the simple thermostats are replaced by true-reading thermostats which transmit the actual temperature to the timeclock.

Optimum time

By continuously monitoring the room temperature, the heating can be turned on earlier when the ambient temperature is low, or later when it is higher. By 'remembering' the rate it was able to heat the house on previous days, the timeclock/controller can determine the optimum time to turn on the heating, giving maximum comfort at minimum cost.

Such a system can show energy savings of up to 20%. By adding an external termometer or combining the system with weather monitoring—broadcast via radio or power distirbution network—considerable further savings are possible.

Celect, developed by a consortium of 25 electricity untilities and manufacturers, led by EA Technology[2] is one such system. It uses a mixture of storage and direct-acting heating, directly influenced by radio broadcast weather data to provide exact programmed temperatures.

Room temperatures and occupancy times are set on a room-by-room basis using a central controller. An optimising controller in the heater calculates the minimum cost for satisfactory comfort requirements based on the day's cost-reflective messages, taken every half hour, and the Radio Teleswitch system weather forecast.

Minimizing costs

For the electricity supplier, the system provides a means of minimizing overall supply and distribution costs by using the novel cost-reflective messages to regulate the daily load pattern.

The EA Technology system is capable of extension to many other applications. The Celect system uses an echelon chip[3] to implement the application and for the required communications within the building power-line signalling is used, eliminating the need for dedicated signal wiring. By furnishing a protocol and an integrated-circuit among its equipment, the company ensures a simple-to-use facility.

Most intelligent home systems use 'modes' to indicate the type of programme being run at any particular time. A mode is simply set using meaningful names and the system does the rest.

For instance, in 'night' mode, the

system will set the heating and hot water to minimum values and, perhaps, set the downstairs security. When the occupant leaves the house, modes such as 'at work', 'at school' or 'shopping' can be selected. The system will automatically set the security, turn off the lights, turn down the heating, and so on.

Learning from actions

By learning from past actions, the system will know when the house will be reoccupied and turn up the heating and switch on other appliances. Almost every household function can be included in a full intelligent system.

Most of the available home automation products are based on systems linked to the existing mains electricity wiring. However, owing to the different voltages in different countries, some sort of standard is considered necessary to make such systems fully worthwhile. To this end, there are several consortia, each endeavouring to set up a standard, while others are engaged in finding an interface to make the different standards compatible.

In contrast with this are organizations which aim to provide interactive services. Companies such as British Telecom, for example, which could supply such a service via its existing telephone lines. Cable TV, satellite companies and even network-linked PCs could all be considered for the interactive services complex.

Available wiring

One consortium which boasts a 'plug and play' capability is the European Home Systems Association (HS). The control signals for the system run on readily available wiring: electric power wiring, TV coaxial cable or the telephone twisted pair.

Features of the Home Systems approach include modularity and ease of growth. It is a fully open system, so new products developed within the system's framework will work alongside the old. For instance, Phillips APG[4] has introduced its HS-Tools Developer's Workstation which allows the creation of remotely controlled 'plug and play' products which automatically cooperate. To extend and upgrade the system, more devices and feature controllers can always be added.

Another consortium is the European Installation Bus Association (EIBA). The EIB system employs just one 'bus' (line) along which all devices communicate. Consequently, much less wiring is required, thus simplifying installation. The bus constitutes a direct connection and controls all functions without the need for a control centre.

One British member company in this group is Home Automation Ltd[5]. This company has recently announced its D-bus concept, a building control product range. The range includes passive infra-red coded transmitters and non-intelligent interfaces. The bus is a twisted pair of wires that interconnect the accessories.

Conventional switch controls, such as thermostats, timers and wall switches can be incoroprated in the installation via connection to a D-bus signal collector.

Serial bus

An alternative low-cost home automation system is offered by Skylake Talix[6]. This company's serial bus uses a serial applications interface—a proprietary ASIC (application-specific integrated circuit)—to provide the link from the processor module to as many as 15 remote modules.

The ability of an integrated home system to coordinate individual subsystems will prove a great advantage for the disabled person. A project aimed at providing such services is underway at the University of Reading[7]. Known as the HS-ADAPT (Home Systems - Access to this Technology for Disabled and Elderly People), it is supported by the European Commission's TIDE programme and is based on the HS specification.

The project is concentrating on the development of a range of portable user interfaces that have bi-directional communications to the rest of the system over an IR link.

Also specifically for the disabled, Hugh Streeper[8] has produced a portable control system. Known as Fox, the controller was awarded the British Institute of Electrical Engineers' 1995 prize for helping the disabled.

A small lightweight device, Fox can be easily mounted on wheelchairs, beds, armchairs and stands. It is radio-frequency operated and can control mains and battery powered equipment. It has 22 control selections, each controlling up to three functions,

Produced and published by **ELEKTOR ELECTRONICS (Publishing)**

Editor: Len Seymour
Technical Editor: Jan Buiting

Editorial & Administrative Offices:
P.O. Box 1414
DORCHESTER DT2 8YH
England
Telephone: (01305) 250995 (*National*)
or +44 1305 250995 (*International*)
Fax: (01305) 250996 (*National*)
or +44 1305 250996 (*International*)

U.K. Advertising Office:
3 Crescent Terrace
CHELTENHAM GL50 3PE
Telephone: (01242) 510760
Fax: (01242) 226626

International Advertising Office:
Elektuur BV
P.O. Box 75
6190 AB BEEK
The Netherlands
Telephone: +31 46 4389 444
Fax: +31 46 437 0161

Subscriptions:
World Wide Subscription Service Ltd.
Unit 4, Gibbs Reed Farm
Pashley Road
TICEHURST TN5 7HE
Telephone: (01580) 200657 (*National*)
or +44 1580 200657 (*International*)
Fax: (01580) 200616 (*National*)
or +44 1580 200616 (*International*)

Head Office:
P.O. Box 75
6190 AB BEEK
The Netherlands
Telephone: +31 46 438 9444
Fax: +31 46 437 0161
Managing Director:
Johan H. Boermann
Deputy Managing Director:
Menno M.J. Landman
Editor-in-Chief/Publisher:
Pierre E.L. Kersemakers
Commercial Manager:
Karel van Noordenne

Distribution:
SEYMOUR
1270 London Road
LONDON SW16 4DH

Printed in the Netherlands by NDB, Zoeterwoude

Overseas editions:
FRANCE
Elektor sarl
Les Trois Tilleuls
B.P. 59; 59850 NIEPPE
Editor: G.C.P. Raedersdorf

GERMANY
Elektor Verlag GmbH
Süsterfeldstr. 25
52072 AACHEN
Editor: E.J.A. Krempelsauer

GREECE
Elektor EPE
Karaiskaki 14
16673 Voula—ATHENA
Editor: E. Xanthoulis

INDIA
Elektor Electronics PVT Ltd
Chhotani Building
52C, Proctor Road, Grant Road (E)
BOMBAY 400 007
Editor: C.R. Chandarana

ISRAEL
Elektorcal
P O Box 41096
TEL AVIV 61410
Publisher: M. Avraham

NETHERLANDS
Elektuur BV
Peter Treckpoelstraat 2–4
6191 VK BEEK
Editor: P.H.M. Baggen

POLAND
Elektor Elektronik
02–777 Warszawa 130
Skrytka Pocztowa 271
Editor: W. Marciniak

PORTUGAL
Ferreira & Bento Lda.
Campo Grande, 56 – 8°/9°
1700 LISBOA
Editor: F. Ferreira de Almeida

SPAIN
Resistor Electronica Aplicada SA
P.. de la Castellana, 212–4° Izda
28046 MADRID
Editor: Agustin Gonzales Buelta

SWEDEN
Electronic Press AB
Box 5505
14105 HUDDINGE
Editor: Bill Cedrum

and ten stored telephone numbers.

With all this activity, HBES technology has passed the development stage, and a good many British companies are rady to take a leading role in the creation of the ultimate in home comfort.

1 British Automated Homes and Buildings Association, 34 Palace Court, London W2 4HY.
2 EA Technology, Capenhurst, Chester United Kingdom CH1 6ES
3 Echelon Europe Ltd, 77 Fulham Palace Road, London, United Kingdom W6 8JA
4 Phillips APG, Cross Oak Lan, Redhill, Surrey, United Kingdom RH1 5HA
5 Home Automation, Bumpers Way, Chippenham, Wiltshire,United Kingdom SN14 6LF
6 Skylake Talix, Unit 3, Lulworth Business Centre, Nutwood Way, Totton, Hampshire, United Kingdom SO4 3WW
7 University of Reading, Department of Cybernetics, Whiteknights, Reading, Berkshire, United Kingdom RG6 2AY
8 Hugh Streeper Ltd, Queen Mary's University, Roehampon Disabled Centre, Roehampton Lane, London, United Kingdom SW15 5PL

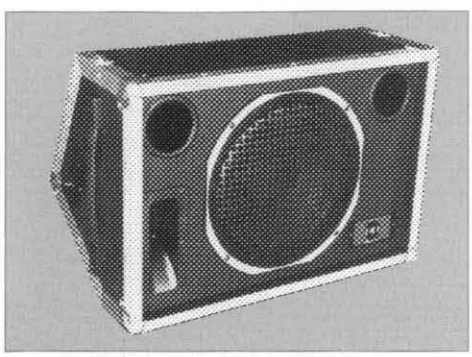
MORRISON (MICROS) MOVE

Due to the steady growth of business over th past few years, Morrison (Micros) has decided to set up in Malta under the name *Leading Edge Technology Ltd.* The company will continue with its present product range, but will be capable of a much higher production capacity than at present, and will have more facilities for research and development. This will ensure that it will be able to provide innovative new products and grow with the ever changing electronics market.

Leading Edge Technology Ltd, White Rose House, Trio IX - Xintill, Tarxien, Malta PLA11. Telephone/fax +356 678 509.

ibl FC12-200M FROM B.K. ELECTRONICS

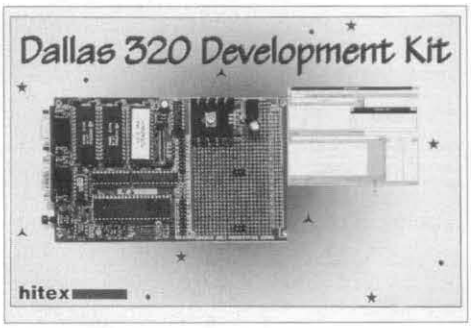
To complement their range of loudspeakers, Infinite Baffle Limited have introduced a foldback monitor stage loudspeaker. The loudspeaker cabinet has been designed so that the user has a choice of two angles, 30° or 60° to suit most applications. Bass response has been tailored especially for foldback applications. Overall response extends down to 45 Hz and up to 20 kHz.



The cabinet comes fully flight cased and has a steel grille fitted on the front to protect the 12" driver. It measures 590×400×357 mm (w×h×d) and weighs 16 kg.

The unit is competitively priced at £ 105, incl. VAT, each plus £ 12·50 p&p, and is available only from B.K. Electronics, Unit 1, Comet Way, Southend-on-Sea, Essex, United Kingdom SS3 9NR. Telephone +44 (0)1702 527 572. Fax +44 (0)1702 420 243.

NEW FROM HITEX



Hitex have available a low-cost development kit, Kit-320, for the Dallas 80C320 high-performance microcontroller. The kit is a complete tool chain and consists of the Keil assembler and ANSI C compiler with optimized libraries for the 80C320, a prototyping board with the unique ***HiTOP*** monitor debugger and all cables and power supplies.

To help designers get through the first stage of a network project, Hitex have a development board and kit for the popular Controller Area Network, CAN, protocol. The board is available on its own or as part of a kit with C compiler, assembler, ***HiTOP*** monitor and comprehensive demo code. The monitor has C source level debugging and special user pages which allow an engineer to directly examine and modify the state of each CAN mode.

Hitex have recently introduced a novel form of 68000 target monitor to complement their BDM debugger. The ***telemon68K*** Universal uses a technique called **ROMLINK** which connects directly to the EPROM socket of the 68000 target and communicates to he CPU via the processor bus. The connection to the PC is a parallel link to the printer port which gives fast communication to the target. The **ROMLINK** adaptor needs little configuration and sits in the boot EPROM socket of the target hardware.

Hitex have many more products available: further information from Hitex (UK) Ltd, Warwick University Science Park, Coventry, United Kingdom CV4 7EZ. Telephone +44 (0)1203 692 066. Fax +44 (0)1203 692 131.

DIGITAL SIGNAL PROCESSING THE ROADSHOW

Miller Freeman Technical (formerly Morgan Grampian Technical) have designed a series of specialist events focusing on digital signal processing. Providing an educative and a purchasing forum for technical management and engineers, Digital Signal Processing – The Roadshow will be visiting a venue near to you from October 1995 to September 1996. It features a technical conference, a free exhibition and free supplier product presentations.

Glasgow	24–25 October 1995
Bristol	7–8 November 1995
Cambridge	14–15 February 1996
Birmingham	17–18 April 1996
Manchester	5–6 June 1996
London	25–26 September 1996

Full details from Miller Freeman Technical Ltd, 6 Faraday Close, Clacton-on-Sea, Essex, United Kingdom CO15 4BR. Telephone +44 (0)181 614 8042, or surf the net on http://www. digii Net.co.uk/

TEST '95 EXHIBITION & CONFERENCE

The Test '95 Exhibition & Conference, sponsred by *TEST* magazine, the industry's European journal, will take place at the Telford International Centre on 10–11 October 1995. The underlying theme this year will be innovation, with subjects ranging from the use of a PC for low-cost ATE to advanced manufacturing test, from device characterization to environmental testing. There will also be an update on the VXI plug & play initiative, together with an open forum with the VXI vendors – a session that promises a very lively debate. Details from:

Angel Business Communications Ltd, Kingsland House, 361/373 City Road, London, United Kingdom EC1V 1LR. Telephone +44 (0)171 417 7400. Fax +44 (0)171 417 7500.

SIMPLE CAPACITANCE METER

Based on an idea of Dr K.C. Rohwer

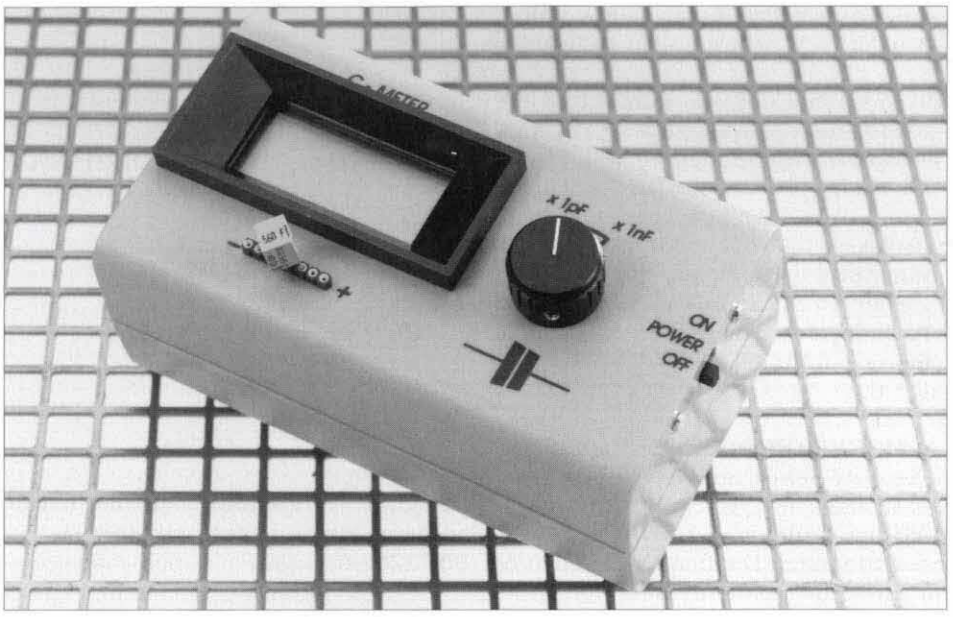
Based on a simple measurement principle and an inexpensive digital voltmeter (DVM) module, a simple capacitance meter is described that can measure, with better than average accuracy, capacitances from 1 pF to 1 µF in two ranges

The capacitance meter is based on the CMOS version of the ubiquitous Type 555 timer IC. As shown in **Fig. 1**, it is arranged as an astable multivibrator (AMV), whose frequency is controlled by the unknown capacitor, C_X. The output frequency of the AMV is inversely proportional to the measured capacitance. In other words, the AMV functions as a capacitance-to-frequency (C/f) converter.

The AMV is followed by a frequency-to-voltage (f/U) converter, whose output voltage is inversely proportional to the frequency of the AMV. This means that the voltage measured by the digital-voltmeter (DVM) module can be read on the liquid-crystal display (LCD) directly as a capacitance.

Note that it is possible to use a stand-alone DVM instead of the built-in DVM module.

The block U_{ref} in Fig. 1 provides a reference voltage for both the AMV and the f/U converter.



The AMV

The AMV is based on the standard oscillator circuit formed by IC_1 in **Fig. 3**. The frequency of the oscillator is determined by the *RC* network at its pins 2 and 6. The circuit has two measurement ranges, selected by S_1. Depending on the position of the switch, capacitor C_2 or C_3 is connected in parallel with the unknown capacitance, C_X. These capacitors compensate for the effect of parasitic capacitances; this will be reverted to later on.

In both measuring positions of S_1, capacitance C_X is charged to the 3.3 V reference voltage via two identical resistors. When C_X is being charged, pin 7 of IC_1 is high-impedance. This changes, however, as soon as the voltage across the capacitor reaches 2/3 of the reference voltage (= operating voltage of the IC). A switch in the IC then connects pin 7 to ground, whereupon C_X is discharged via R_1 or R_2, depending on the position of S_1. When the potential across C_X has dropped to 1/3 of the reference voltage, the switch in IC_1 removes pin 7 from ground, whereupon this becomes high impedance again. A new charging cycle then begins. This process repeats itself at a frequency determined by the value of C_X. Since a small capacitance is charged and discharged more quickly than a large one, the frequency with a small capacitance is higher than with a large one.

At each transit from charging to discharging, the logic level at the output of IC_1 changes. This means that the AMV generates a rectangular signal, whose frequency diminishes when C_X increases. Since IC_1 is a CMOS type, the level of its rectangular output signal readily reaches the level of the supply voltage when the output load is small. The supply voltage is derived from the reference voltage via IC_{2a} and IC_3 and is thus very stable.

As mentioned earlier, capacitors C_2 and C_3 in parallel with C_X compensate for the effect of parasitic capacitances. It would have been simple to choose a 15 nF trimmer for C_3, were it not for the fact that such high-value trimmers do not exist. Therefore, a fixed capacitor of a value some 1000 greater than the parasitic capacitance is used. In the lower range, C_2 is set for a value of parallel capacitance commensurate with the measuring range. This ensures that, as a percentage, the capacitive offset contained in the measurement results for both ranges is the same, provided there is no further adjustment with P_1 when the other measurement range is selected. Preset P_1 is used to set the display to zero when the C_X terminals are open-circuit; when this is done, the offset voltages of the op amps, and particularly that of IC_{2d}, are set to zero at the same time.

f/U converter

In contrast to the straightforward design of the AMV, the function of the f/U converter is rather more complex. It consists of:

- an integrator with automatic offset

Brief specification

Measuring ranges	1 pf – 1 nF
	1 nF – 1 µF
Accuracy	Better than 2%
Measuring frequency	200 Hz – 10 kHz
Operating voltage	6.6 – 12 V
Current drain	4 mA (measuring circuit)
	1 mA (DVM-LCD)

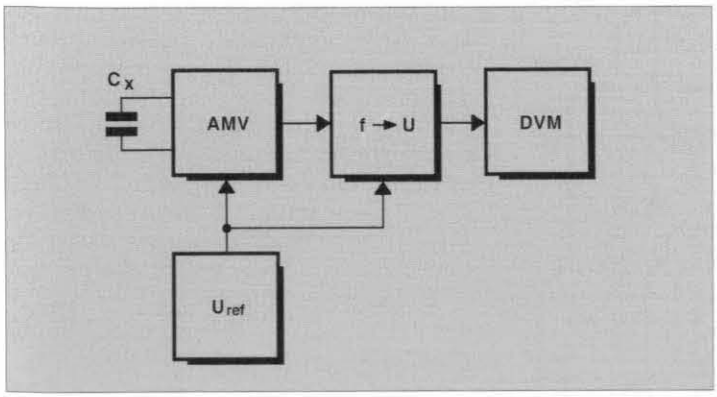


Fig. 1. Block diagram of the capacitance meter with integral digital-voltmeter module.

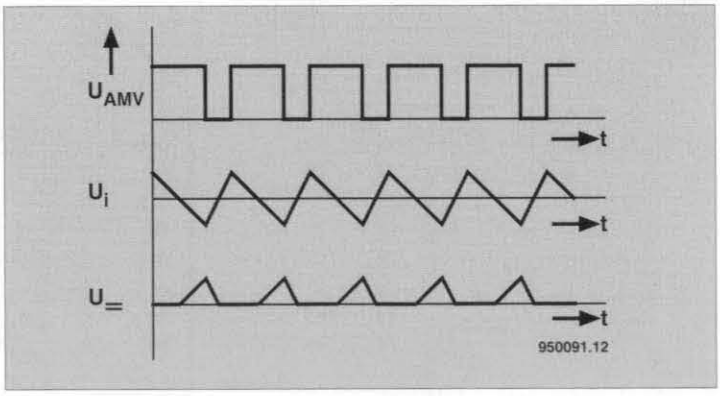


Fig. 2. Output signal waveforms of the AMV, integrator and half-wave rectifier.

correction;
- a half-wave rectifier;
- an averaging circuit;
- a direct-coupled amplifier.

Any direct voltage on the timer output is prevented from reaching the f/U converter by C_4.

The supply for the converter is taken from the reference voltage, which thus forms the zero potential of the rectangular signal at pin 6 of IC_{2b}: U_{AMV} in **Fig. 2**.

The output signal of IC_{2b} is the centre waveform in Fig. 2, U_i. This is a triangular signal whose amplitude diminishes with rising frequency (as in a low-pass filter). Owing to the automatic offset correction, the triangular signal is symmetric w.r.t. the base line (= 3.3 V reference voltage).

Half-wave rectifier IC_{2c} removes virtually all of the negative half cycles. The resulting signal is inverted at the output of the rectifier, resulting in the lower waveform, $U_=$ of Fig. 2.

The averaging circuit consists of low-pass filter R_{11}-C_7, which is terminated into a high-impedance load by d.c. amplifier IC_{1d}. The op amp matches the voltage level and range to that of the DVM module (or DVM). The peak voltage, after averaging, is 400 mV at the top end of the measuring ranges (1 nF and 1 µF respectively).

Reference-voltage source

Measurements are linearly related to the reference voltage, U_r. It is for that reason that the source of this voltage requires careful design. Apart from being independent of the battery voltage, it must be very stable over a wide temperature range. Also, because of the cost of 9 V batteries, the necessary voltage regulator should draw a very small current only. Consequently, a low-power voltage-reference IC is used: IC_3. Unfortunately, this can provide an

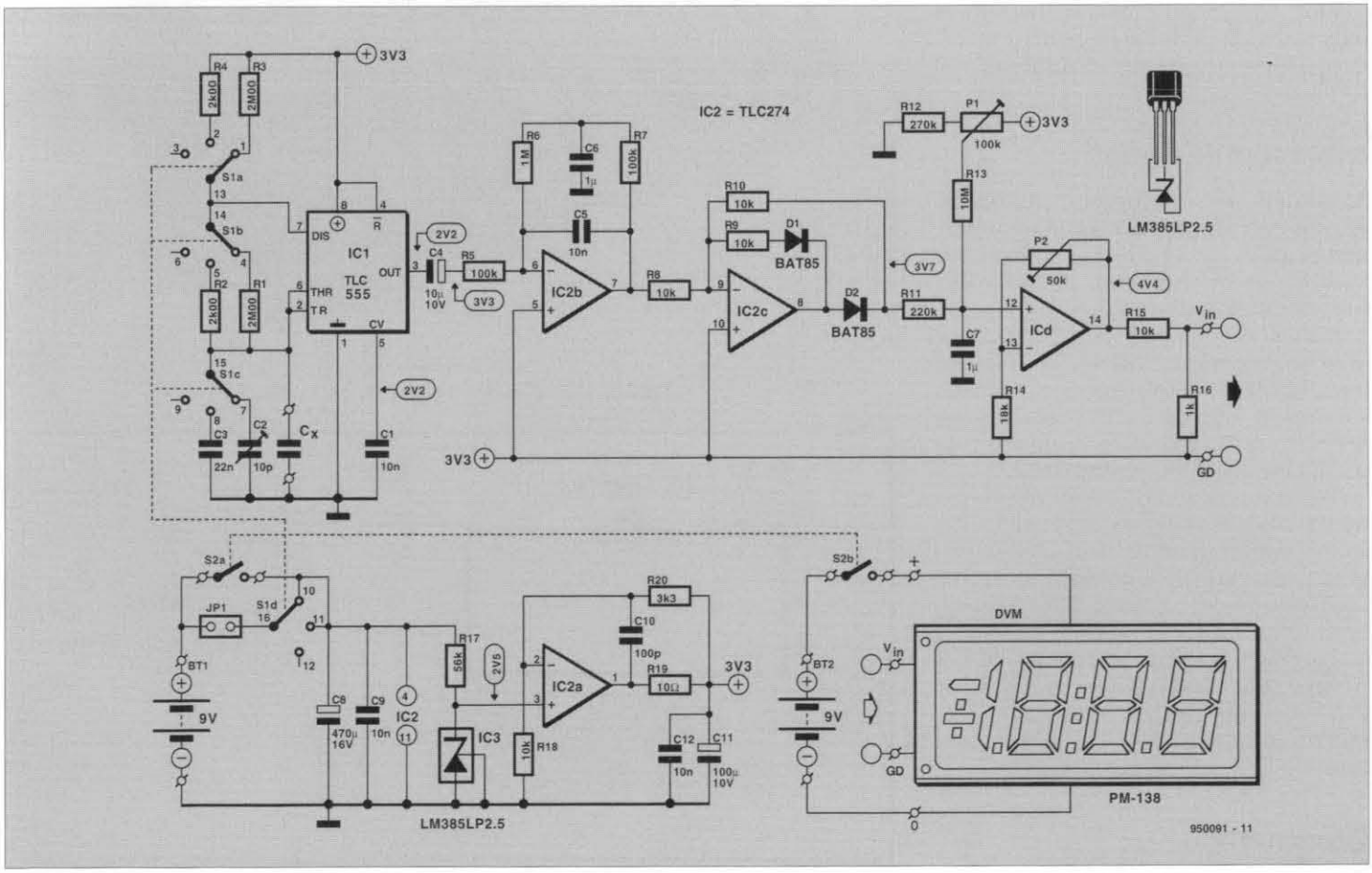


Fig. 3. Circuit diagram of the capacitance meter with integral digital-voltmeter module.

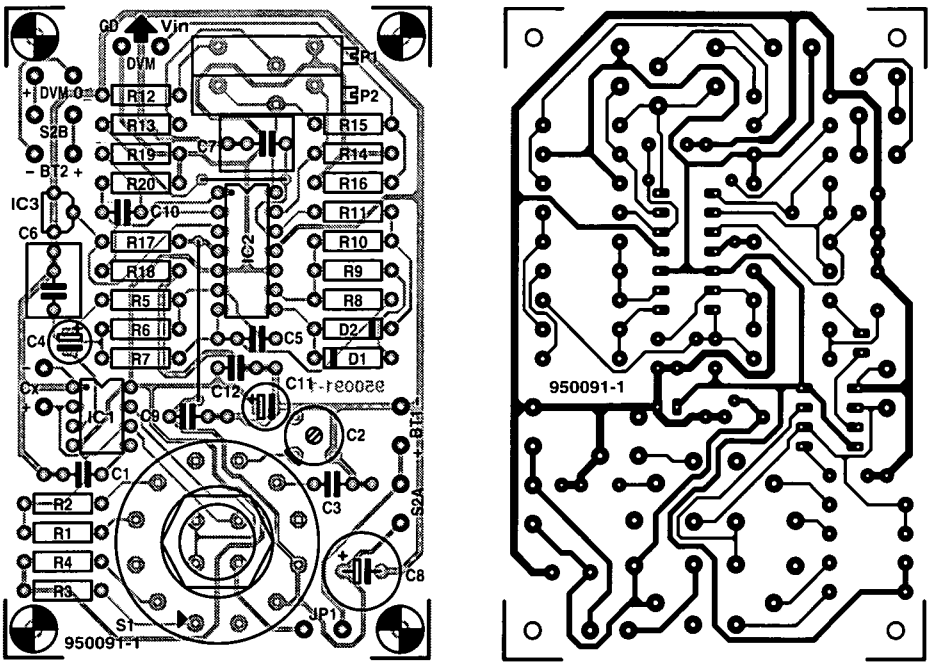


Fig. 4. Printed-circuit board for the capacitance meter.

output of not more than 2.5 V. Since buffering is required in any case, this voltage can be raised to some extent by buffer IC_{2a}. In this way, a value of 3.3 V is obtained, which is a good compromise between low battery drain and good stability. This level is just under half the battery voltage.

Capacitors C_{11} and C_{12} at the output of the voltage source enhance the decoupling at high frequencies. The resulting capacitive load at the output of the op amp is reduced by R_{19}. Moreover, C_{10} improves the stability in the feedback loop.

DVM module or not?

As stated earlier, the circuit can be used as an adaptor for a digital multimeter (2 V direct voltage range), or with a permanently built-in DVM module with a 200 mV range.

When the circuit is used as an adaptor, its supply voltage is switched on and off with the rotary switch on the board. Switch S_2 and resistor R_{16} are then not used: wire bridge JP_1 on the board must be used instead.

When a DVM module is used (which, by the way, is not very expensive), the rotary switch must be restricted to two positions with the end-stop. Since the module needs its own 9 V battery, double-pole switch S_2 then serves as on/off switch. Wire bridge JP_1 is not needed, but R_{16} is: it scales down the 0–2 V output voltage range of the measuring circuit to 0–0.2 V at the DVM input.

Construction

Populating the printed-circuit board shown in **Fig. 4** should not present any undue difficulties, but remember the earlier comments regarding wire bridge JP_1, switch S_2 and resistor R_{16}. The completed board is shown in the photograph in **Fig. 5**. When all is done, the meter is best built into a suitable enclosure for which a suggested front panel is shown in **Fig. 6**. Note that neither the printed-circuit board nor the front-panel overlay are available ready-made.

Calibration & use

After the supply has been switched on, it is always best to wait about 30 seconds to give the integrator time to settle down to a stable state.

With the C_x terminals open, select the upper measuring range (1 nF to 1 µF) and adjust P_1 to obtain zero reading on the display or external DVM. Then, select the lower measuring range (1 pF to 1 nF) and adjust C_2 for zero reading on the display or external

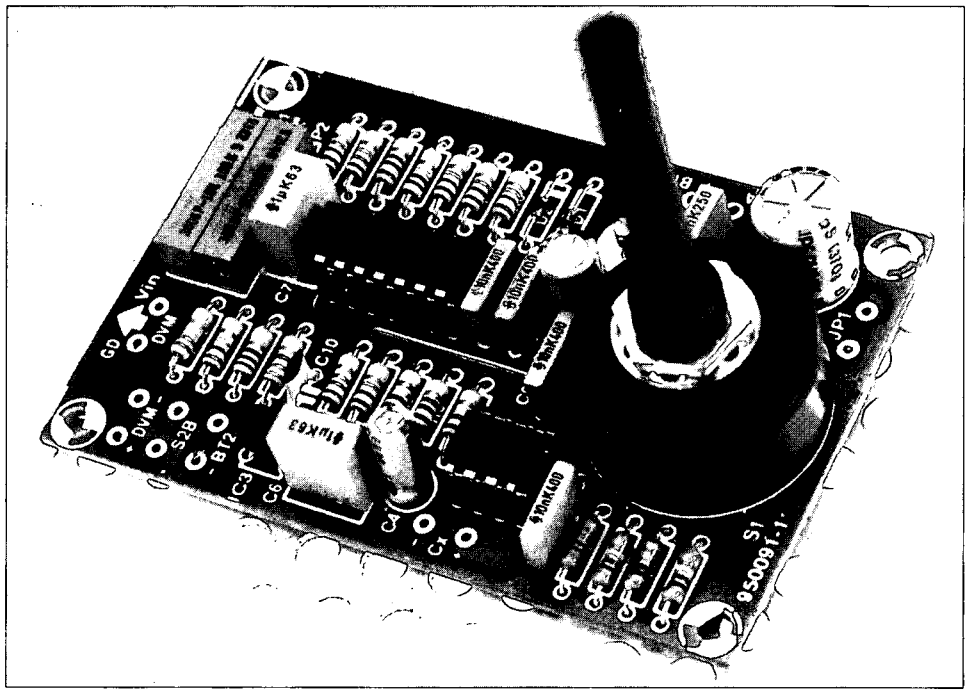


Fig. 5. Completed printed-circuit board.

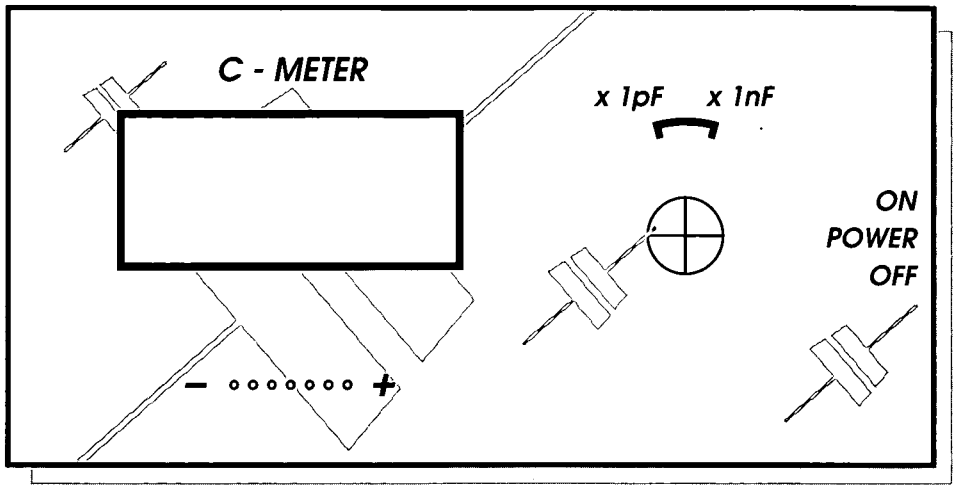


Fig. 6. Suggested front panel layout for the capacitance meter.

Measurement principle

The charging and discharge times, t_{on} and t_{off} respectively, of the capacitor on test, C_x, and thus the pulse and pause times at the output of timer IC_1, are given by

$$t_{on} = 2RC\ln 2$$

and

$$t_{off} = RC\ln 2.$$

The priod of the output signal is then

$$T = t_{on}+t_{off} = 3RC\ln 2,$$

and the frequency of the output signal is:

$$f = 1/3RC\ln 2.$$

Since the operating voltage, U_B, of the timer is equal to the reference voltage, U_r, the output voltage, U_o, providing the load is small, is

$$U_o = U_r.$$

After the direct voltage component has been removed, the signal, U_i, applied to the integrator is

$$U_i = U_r - U_{av},$$

where $U_{av} = (t_{on}/T)U_r$,

so that

$$U_i = (1 - t_{on}/T)U_r.$$

The output voltage of the integrator is

$$u_o = -\frac{1}{R_5C_5}\int u\,dt$$

If $0<t<t_{on}$,

$$u_o = \frac{1}{R_5C_5}(1-\frac{t_{on}}{T})U_r t$$

$$= -\frac{1}{R_5C_5}\cdot\frac{t_{off}}{T}\cdot U_r t$$

The peak value of the output voltage of the integrator is

$$u_o = \frac{1}{R_5C_5}\cdot\frac{t_{off}}{T}U_r t_{on}$$

$$\frac{t_{off}}{T} = \frac{1}{3}$$

$$t_{on} = 2RC\ln 2$$

$$u_o = \frac{2}{3}\cdot\frac{RC\ln 2}{R_5C_5}U_r$$

Owing to the automatic offset correction in the integrator, its output signal is symmetric w.r.t. the base. Consequently, the average output of the integrator is zero. Thus, the peak value of the input to the half-wave rectifier is $u_o/2$. Its output voltage is

$$u_o = \frac{1}{3}\cdot\frac{RC\ln 2}{R_5C_5}U_r$$

During half the period (T_2), the output voltage is positive. The signal is triangular: the area of its waveform is equal to that of half the peak value. Since during a half period there is no signal, the average value of the voltage at the output of the rectifier is equal to a quarter of the peak value:

$$u_o = \frac{RC\ln 2}{12R_5C_5}U_r$$

Thus, since $C = C_x$:

$$u_o = C_x\cdot\frac{R\ln 2}{12R_5C_5}U_r$$

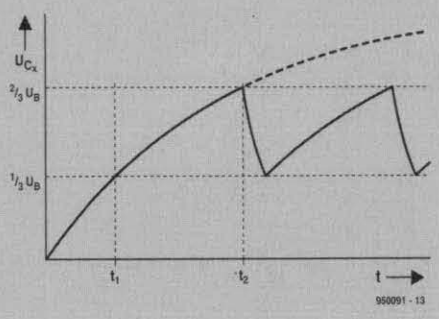
In words: the output voltage of the meter is directly and linearly proportional to the capacitance C_x.

Substituting numerical values in the last equation:

$R_5 = 100\ k\Omega$
$C_5 = 10\ \mu F$
$\ln 2 = 0.693$
$U_r = 3.3\ V$:

$$u_o = 190.62\times C_x\times R,$$

where R is 2 kΩ or 2 MΩ depending on the selected range.



DVM. Note that it takes the display a few seconds to stabilize (owing to time constant R_{11}-C_7).

Next, connect a 1 nF, 1% polystyrene capacitor to the C_x terminals, select the lower measuring range and adjust P_2 to give a reading of 1000 on the display or DMM.

Finally, if at all possible, connect a 1 µF, 1% polystyrene capacitor to the C_x terminals, select the upper measuring range, and adjust P_2 to give a reading of 1000 on the display or DMM.

Parts list

Resistors

R_1, R_3 = 2.00 MΩ, 1%
R_2, R_4 = 2.00 kΩ, 1%
R_5, R_7 = 100 kΩ
R_6 = 1 MΩ
R_8–R_{10}, R_{15} = 10 kΩ
R_{11} = 220 kΩ
R_{12} = 270 kΩ
R_{13} = 10 MΩ
R_{14} = 18 kΩ
R_{16} = 1 kΩ (see text)
R_{17} = 56 kΩ
R_{18} = 10 kΩ
R_{19} = 10 Ω
R_{20} = 3.3 kΩ
P_1 = 100 kΩ multiturn horizontal preset
P_2 = 50 kΩ multiturn horizontal preset

Capacitors:

C_1, C_5, C_9, C_{12} = 10 nF, polypropylene
C_2 = 10 pF trimmer capacitor
C_3 = 22 nF, polypropylene
C_4 = 10 µF, 10 V, vertical
C_6, C_7 = 1 µF, polypropylene
C_8 = 470 µF, 16 V, vertical
C_{10} = 100 pF, ceramic
C_{11} = 100 µF, 10 V

Semiconductors:

D_1, D_2 = BAT85

Integrated circuits:

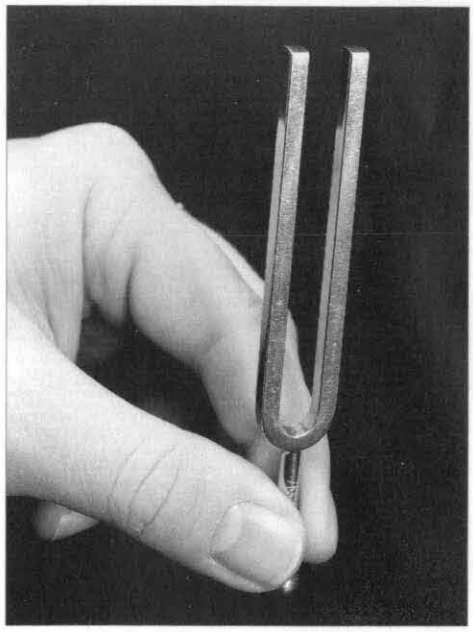
IC_1 = TLC555
IC_2 = TLC274
IC_3 = LM385LP2.5

Miscellaneous:

JP_1 = wire bridge, see text
S_1 = 4-pole, 3-position rotary switch for PCB mounting
S_2 = 2-pole toggle switch
BT_1, Bt_2 = 9-V (PP3) battery with connection clips
DVM = $3^1/_2$ digit LCD DVM for 9 V operation, measuring range 0–199.9 mV (see text)
PCB not available ready made
Front panel foil not available ready made

[950091]

TUNING FORK AMPLIFIER



Design by T. Giesberts

An amplifier is described which, after a tuning fork has been struck, gives a continuous 440 Hz tone (= International A), so that the user has both hands free to tune the relevant instrument.

The setup of the tuning fork amplifier is shown in the block diagram of **Fig. 1**. The operation is comparable to that of a crystal oscillator, in which the crystal has been replaced by a tuning fork and a pick-up coil. The tone produced by the tuning fork is picked up by the coil and maintained by positive feedback in the amplifier, after which it is made audible by the loudspeaker.

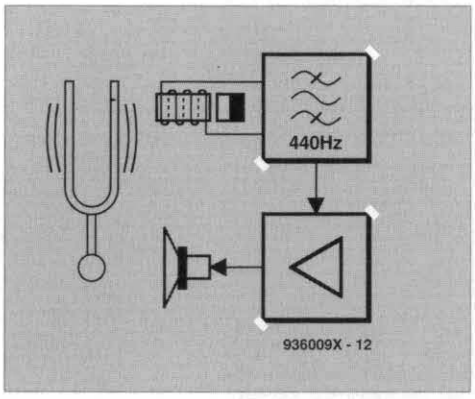


Fig. 1. Block diagram of the tuning fork amplifier.

The pick-up coil is similar to that in an electric guitar. The vibrations of the tuning fork disturb the magnetic field in the core, which causes a small alternating voltage to be induced in the coil.

The coil voltage is applied to a narrow-band 440 Hz filter, which removes the harmonics and other unwanted frequencies. The 440 Hz signal is then applied to an a.f. amplifier, whose output is applied to a small loudspeaker. Note that as long as the fork vibrates, the loudspeaker will sound and as long as the loudspeaker sounds, the fork will vibrate. The circuit thus acts as an oscillator.

Circuit description

The coil is a solenoid from a second-hand relay, preferably a 24 V type because of its greater sensitivity. A small permanent magnet is fixed at the back of the solenoid to create a permanent magnetic field.

The coil potential is applied to 440 Hz filter IC_1. The resistors and capacitors associated with the IC, except decoupling capacitors C_3 and C_4, determine the properties of the filter, such as the central frequency, f_0, the Q(uality) factor and the amplification.

The most important parameter for the present application is the central frequency which must be 440 Hz to comply with the International tuning frequency (note A). Preset P_1 allows f_0 to be varied from 352 Hz to 528 Hz.

The 440 Hz signal is applied to a.f. amplifier IC_2 via C_5. This stage has been designed to give maximum amplification at 440 Hz to ensure further attenuation of any harmonics. The amplification, α, is set with P_2.

The output of IC_1 is applied to 'power' amplifier T_1-T_2. Feedback resistors R_7–R_{10} determine the gain of the amplifier: the amplification factor is $(R_9+R_{10})/R_9 = 430/100 \approx 4$. Since the amplifier can be driven almost to the level of the supply line, the output sig-

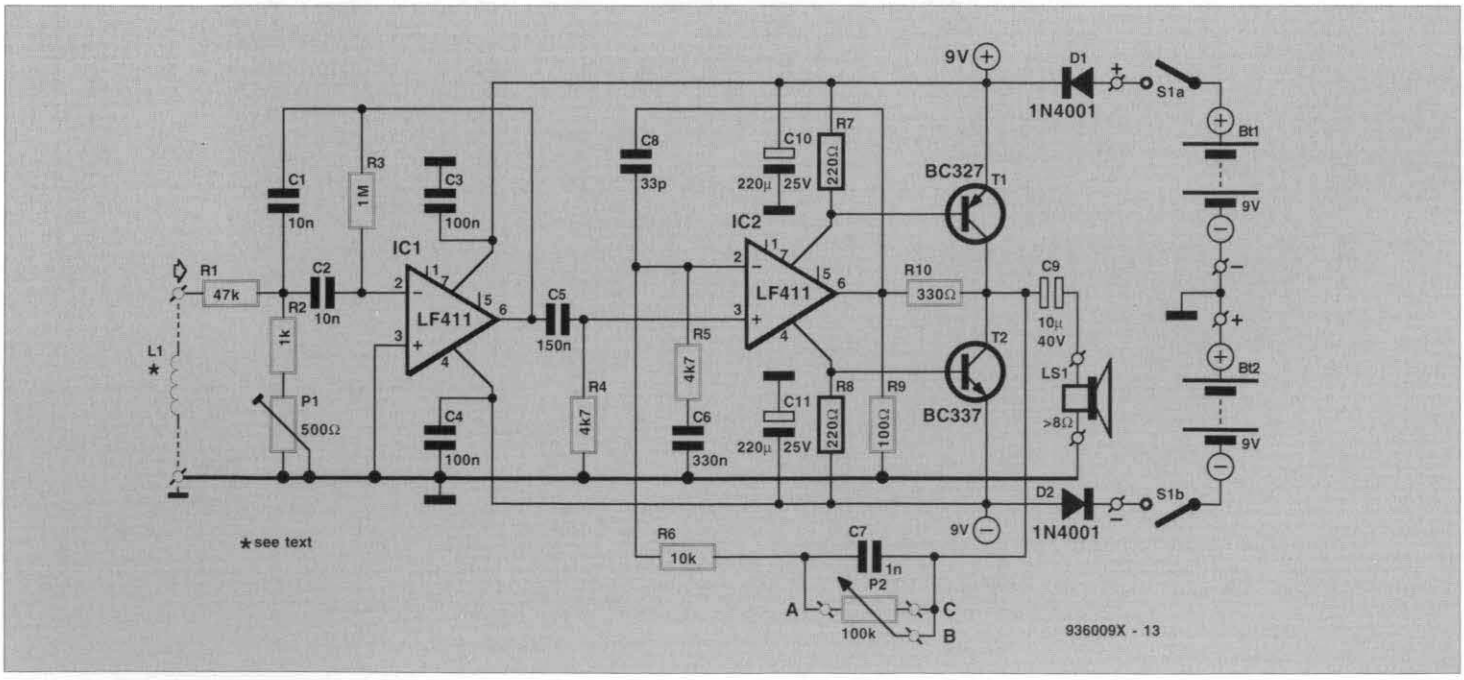


Fig. 2. Circuit diagram of the tuning fork amplifier.

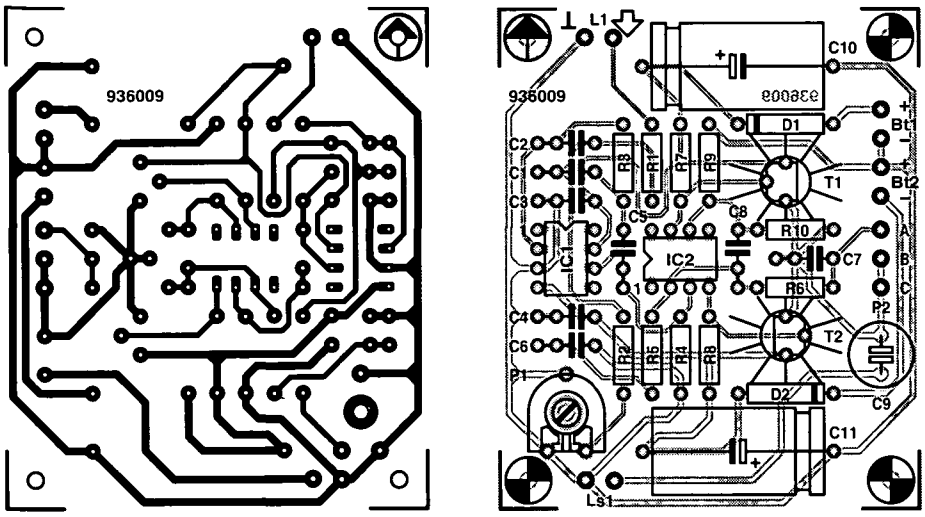


Fig. 3. Printed-circuit board for the tuning fork amplifier.

nal of IC_2 must be not greater than ¼ of the supply voltage.

The power output of the amplifier is applied to the loudspeaker via C_9. To make this capacitor also function as additional impedance to limit the power to the loudspeaker and thus the total current drain, its value has been kept relatively small.

As already stated, the sound waves emanating from the loudspeaker keep the tuning fork in vibration.

Construction

The amplifier is intended to be built on the printed-circuit board shown in **Fig. 3**. In the design of the board, particular attention has been paid to the avoidance of earth loops.

During the building of the board, make sure that enough space is left for the heat sinks of T_1 and T_2: these must not touch any components.

Note that C_{10} and C_{11} must be mounted in opposite directions on the board.

Make sure that IC_1 and IC_2 are placed correctly in their sockets.

In the assembly of the electronics and tuning fork in the enclosure, it is, of course, important that the tuning fork can vibrate freely just as in manual operation. Also, the pick-up coil must be placed close to the tuning fork to avoid undue attenuation of the voltage induced in the coil.

As stated earlier, the pick-up coil is the solenoid removed from a discarded (preferably 24 V) relay. Using superglue, fix a small permanent magnet at the back of the solenoid (where the terminals emerge).

Superglue is also used to fix the loudspeaker and the pick-up coil assembly to the enclosure as shown in **Fig. 4**. For the prototype, perspex was used, but wood is, of course, also perfectly all right. If perspex is used, be careful with the glue, because every spilt drop remains visible.

The board is fixed to the enclosure on spacers as shown in Fig. 4 and **Fig. 5**.

Note that the distance between the tuning fork and pick-up coil should not be greater than 1–1.5 mm. The block in which the tuning fork is clamped may be made from a piece of perspex into which two holes are drilled at right angles. The smaller one is tapped for the fixing screw. Alternatively, it may be

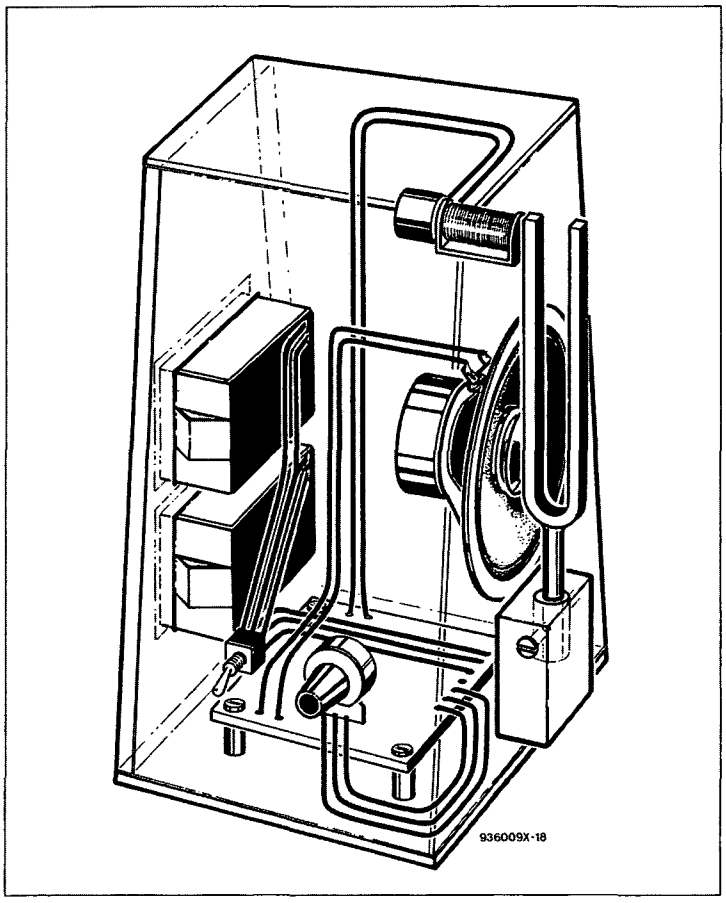


Fig. 4. Artist's impression of the completed amplifier in its enclosure.

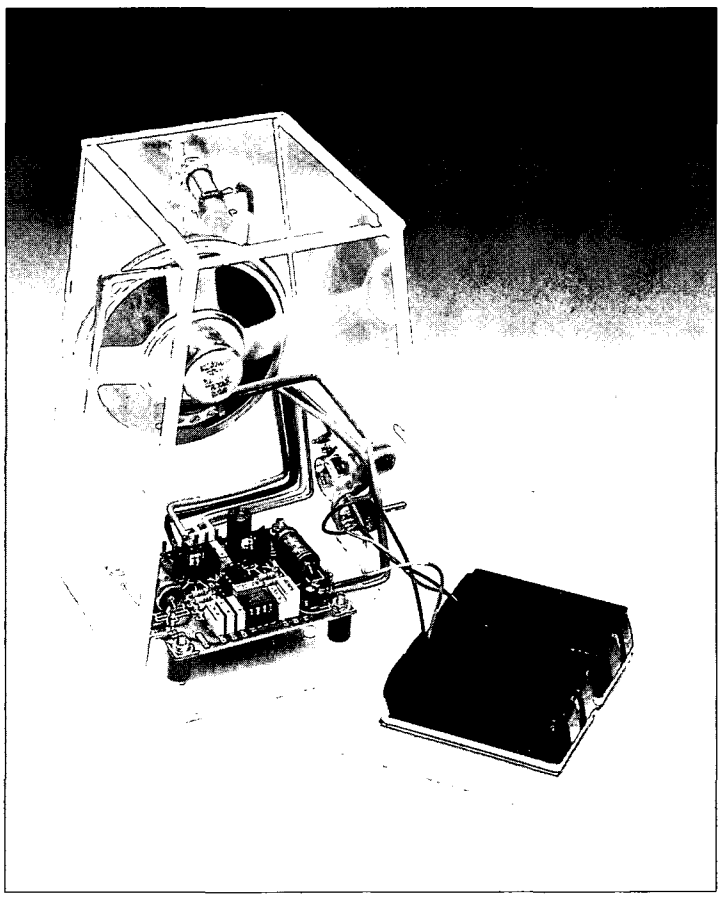


Fig. 5. Photograph of the completed amplifier with the rear panel removed.

made from two pieces of wood as shown in **Fig. 6**. Whatever material is used, the block is glued on to the enclosure (superglue in the case of perspex, good woodglue in the case of wood).

The batteries are best fixed in place with double-sided sticky tape.

Setting up

Switch on the supply, set P_2 to the centre of its travel, and lightly tap the tuning fork. If all is well, a tone at or near 440 Hz will be heard from the loudspeaker. If nothing is heard, turn P_2 for full volume and try again. If this produces no results either, switch off and inspect all connections and the board again.

The unit may be set to exactly 440 Hz (which is a must for musicians) with the aid of a frequency meter. The frequency is set with P_1.

Note that if a tuning fork of different frequency is used, the values of the components in the low-pass filter and the various RC networks must be recalculated.

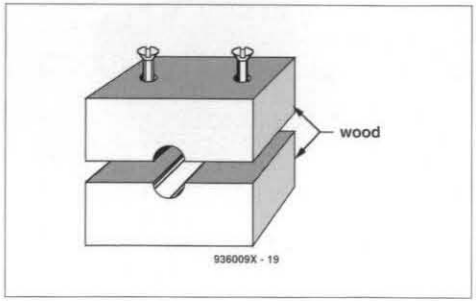


Fig. 6. The clamp for the tuning fork may be made from two blocks of wood.

Parts list

Resistors:
R_1 = 47 kΩ
R_2 = 1 kΩ
R_3 = 1 MΩ
R_4, R_5 = 4.7 kΩ
R_6 = 10 kΩ
R_7, R_8 = 220 Ω
R_9 = 100 Ω
R_{10} = 330 Ω
P_1 = 500 Ω (470 Ω) preset
P_2 = 100 kΩ, linear

Capacitors:
C_1, C_2 = 10 nF
C_3, C_4 = 100 nF
C_5 = 150 nF
C_6 = 330 nF
C_7 = 1 nF
C_8 = 33 pF
C_9 = 10 µF, 40 V, bipolar, radial
C_{10}, C_{11} = 220 µF, 25 V

Semiconductors:
D_1, D_2 = 1N4001
T_1 = BC327
T_2 = BC337

Integrated circuits:
IC_1, IC_2 = LF411

Miscellaneous:
L_1 = see text
Ls_1 = loudspeaker ≥8 Ω, 0.5 W
Bt_1, Bt_2 = 9 V (PP3) battery with connecting leads
2 off heat sinks for T_1, T_2
PCB not available ready made
[936009]

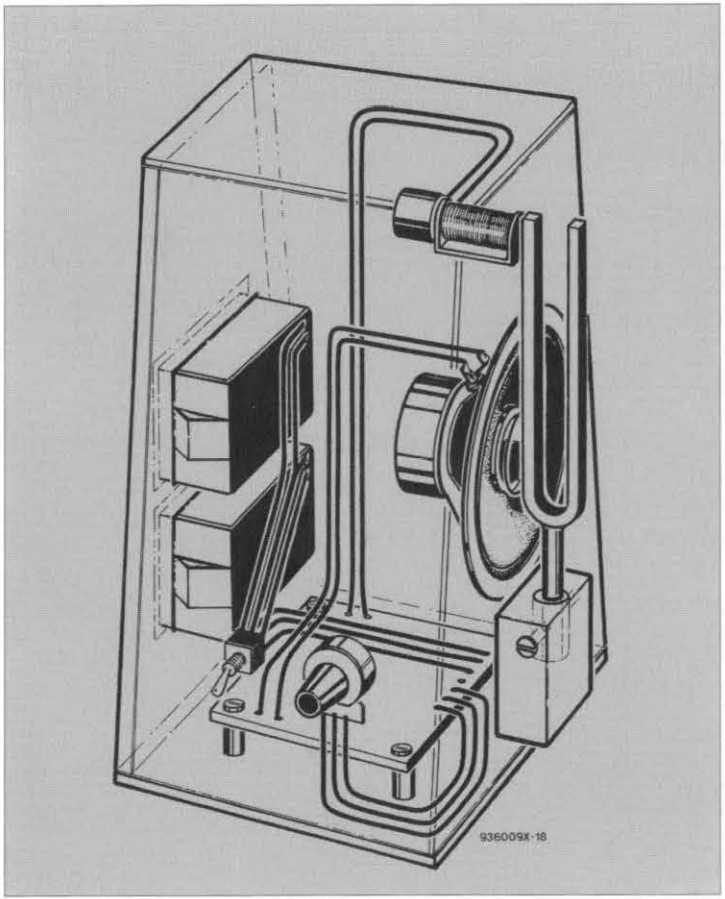


Fig. 4. Artist's impression of the completed amplifier in its enclosure.

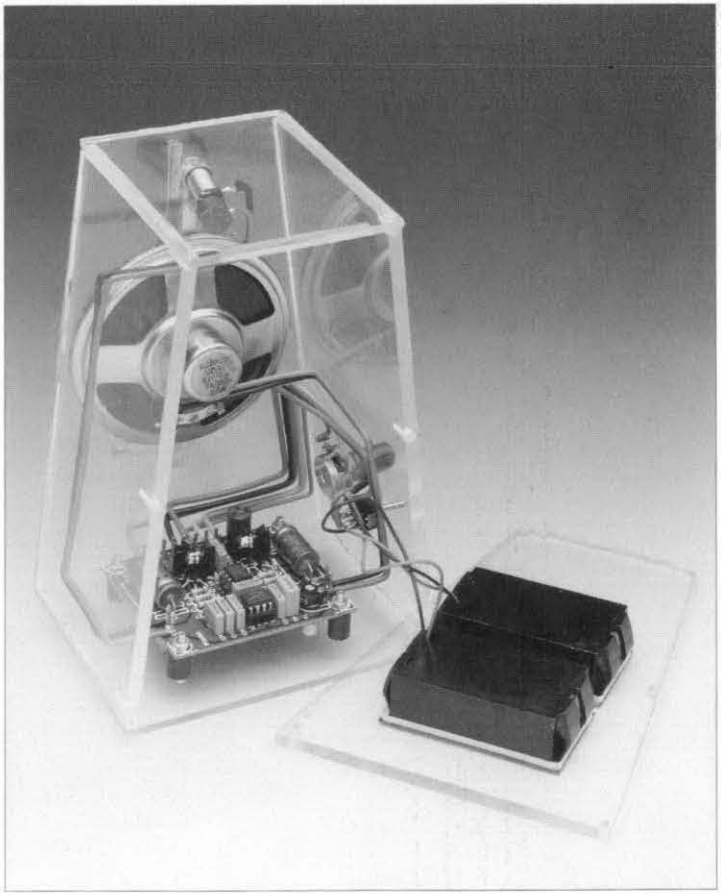
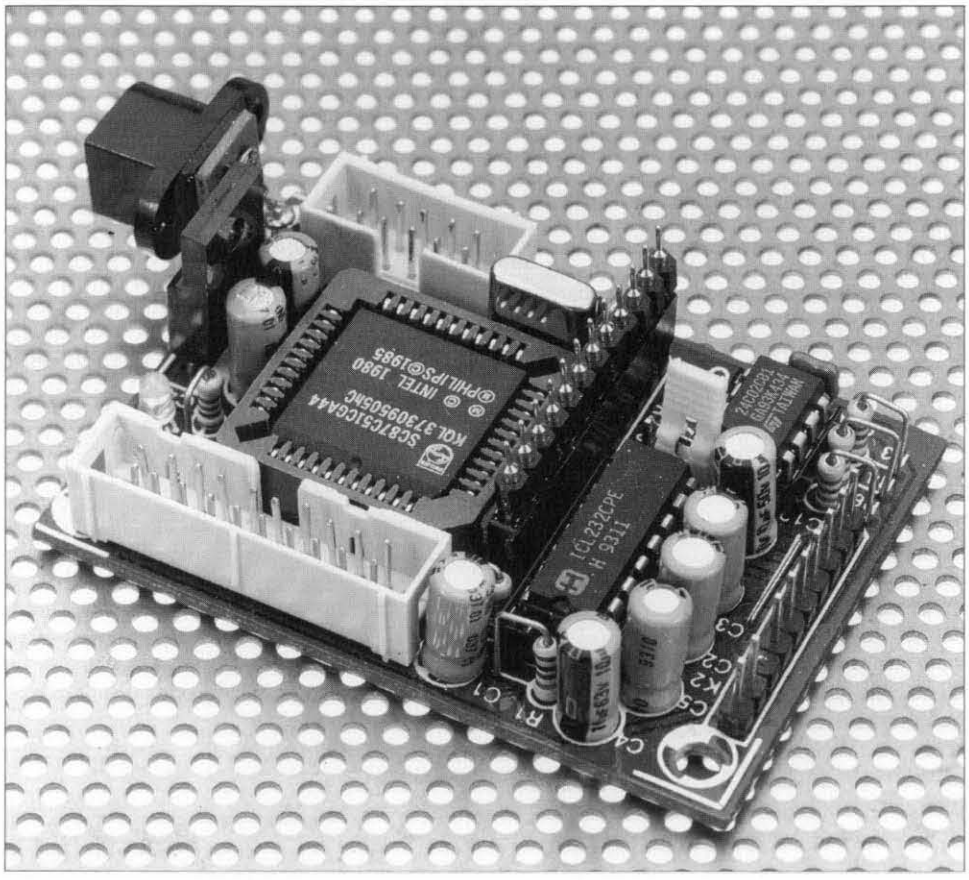


Fig. 5. Photograph of the completed amplifier with the rear panel removed.

'MATCHBOX' BASIC COMPUTER (Part 1)

The miniature size computer described in this short series of articles heralds a new era in DIY microcontroller development systems. The MatchBox computer (MBC) is cheap, simple to program in a BASIC dialect, and uses an EEPROM for non-volatile program storage. Programs for the MBC may be written, edited, debugged and downloaded with the aid of an MS-DOS PC, using the serial RS232 port. To support the hardware, a variety of programming examples 'from the ground up' and system utilities are available on a floppy disk.



Hardware design and software by Dr. M. Ohsmann

MATCHBOX BASIC COMPUTER

- » Easy programming in BASIC-like language
- » Ideal project for newcomers to microcontroller technology
- » Simple 8051 based hardware
- » Flexible memory expansion
- » LCD and RS232 support
- » I^2C bus support
- » EEPROM for program storage
- » No EPROM programmer or emulator necessary
- » Inexpensive development software running on a PC
- » 27 I/O lines
- » 16-byte variables memory in RAM (expandable with I^2C RAMs)
- » EEPROM variables for permanent data storage
- » Low current consumption
- » Compiler, communication/download utility and programming examples available on diskette
- » Short programming course in instalments
- » PCB, disk and controller available through Readers Services

THE great thing about using an EEPROM (electrically erasable PROM) in the MatchBox BASIC computer is that you do not need an expensive EPROM programmer, and do not waste time on erasing and reprogramming a conventional EPROM, which normally takes at least half an hour (even if, say, five bits are wrong on a total of 4,000). After downloading the user program into the EEPROM, the Matchbox computer is fully programmed, and the code is retained for ever, even if the power is switched off. This programming sequence can be repeated up to 100,000 times.

Programs for the MBC originate from a PC, which represents an inexpensive and widely available tool for this function. The MBC requires neither assembly code programming nor an expensive (cross) compiler. Thanks to the complete integration of RS232 (V24), LCD and I^2C drivers in the control software, the MatchBox computer has many applications, and is simple to extend. It is, therefore, eminently suited to control applications. Because of its small size and low current consumption, the MBC may be incorporated into existing equipment which requires any degree of 'intelligence'.

The concept

All the problems, loss of time and money, normally associated with traditional EPROMs used in microcontroller development systems are virtually eliminated by an EEPROM which can be re-programmed while it remains in the circuit. The whole operation of refreshing an entire program takes seconds rather than minutes, and you do not need a UV chip eraser unit, an

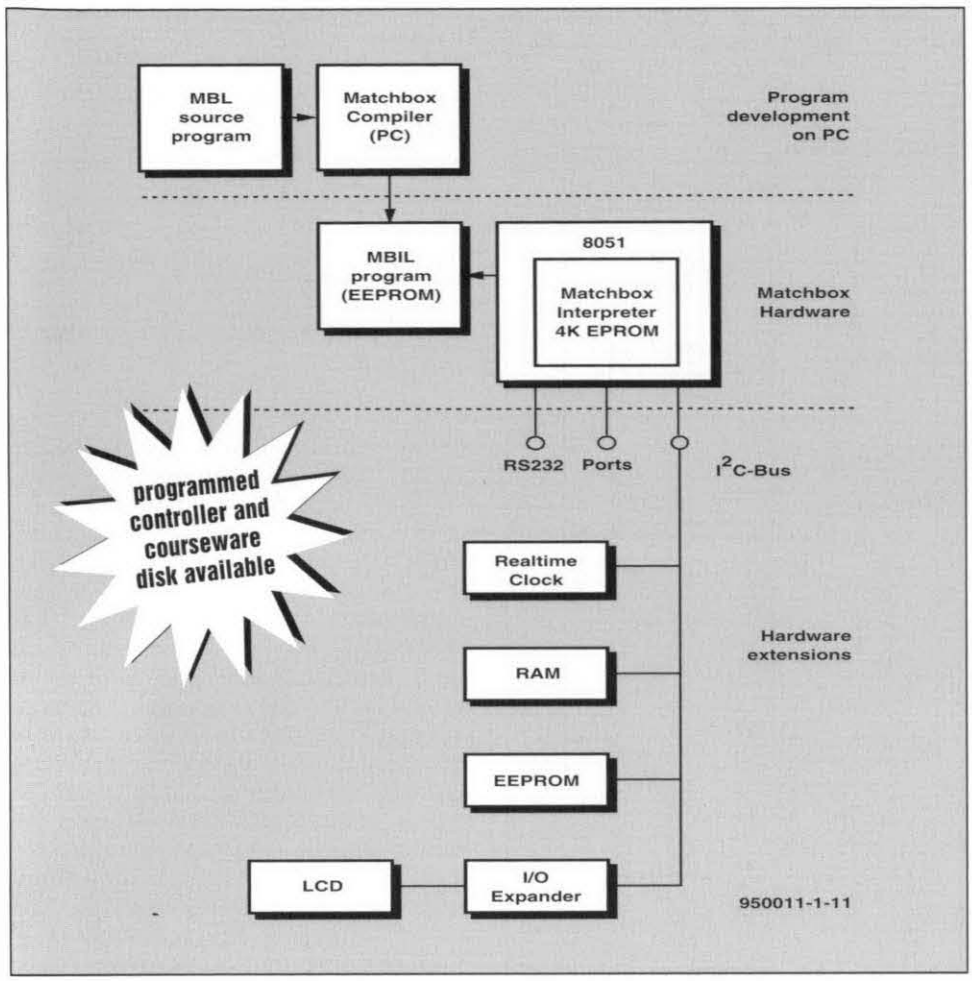


Fig. 1. Programming and hardware overview of the BASIC MatchBox computer project.

MatchBox: hardware and software

The MatchBox computer will be presented to you in greater detail in the next three or four issues of this magazine. This first instalment deals with elementary descriptions of the possibilities of the MatchBox computer, as well as with the hardware (construction). The next instalments will concentrate on the BASIC language used to program the computer.

The programming examples to be discussed have a clear didactic aim, and so turn the instalments into a short course. Small, experimental, projects will be presented which use the MatchBox computer as a starting point.

A collection of useful programs for the MatchBox computer is available on a 3.5-inch MS-DOS diskette, which contains

- **MatchBox Compiler (DOS) including manual.**
- **Over 20 programming examples.**
- **MatchBox menu control (DOS).**
- **Installation instructions.**

order code: 956009-1

EPROM programmer and/or EPROM emulator to get going.

The heart of the MatchBox computer is an 8051-derivate microcontroller. To avoid the use of an external address latch, a serial I^2C EEPROM is used instead of the traditional 28C256 EEPROM. This leaves many port functions offered by the microcontroller free for user applications. These days, serial EEPROMs are available with a storage capacity of up to 8 KBytes, which can hold complex application programs. In the present circuit, however, a 256 or 512 byte EEPROM is used, whose capacity is ample for many small applications.

The processor requires a kind of mini operating system to be able to fetch and execute the user program stored in EEPROM. This operating system is burned into the processor's internal 4-KByte OTP EPROM. The ready-programmed processor is available through our Readers' Services. Apart from allowing programs to be stored into the EEPROM, the operating system also provides an interpreter which translates the tokenized code contained in the EEPROM into machine code. The tokenized code is written in a format called MBIL (MatchBox Interpreter Language), which allows the user program to be stored in compressed form in the EEPROM.

That brings us to the subject of programming. The user writes his/her program in a 'higher programming language', called MBL (MatchBox Language). This text is fed to the MatchBox Compiler (MBC) which (still running on a PC) turns the text into the MBIL format for downloading to the MatchBox computer. These steps are illustrated in **Fig. 1**. Actually, the MBIL (download) format is invisible to the user, so you need not bother about it. As far as you, the user, are concerned, the MatchBox computer understands the MatchBox language (i.e., a dialect of BASIC) directly.

Apart from the hardware discussed in this article, you only need the MatchBox diskette, a PC and an RS232 cable to create an astoundingly simple, yet powerful, microcontroller development system.

Circuit description

The complete circuit of the MatchBox computer is shown in **Fig. 2**. The main component is CPU IC_1, which contains the MatchBox interpreter in its internal ROM. The well known MAX232 (IC_2) is used as a two-way level converter between the circuit and the RS232 lines to the PC's serial port. An I^2C bus is created with the aid of CPU pins P3.4 (T0; I^2C SCL) and P3.5 (T1; I^2C SDA). This bus forms the connection to the external program memory, an EEPROM (IC_3). To enable I^2C devices to be connected which are capable of generating an interrupt (for example, a real-time clock, or an I/O expander), pin P3.3 ($\overline{INT1}$) is employed, and also taken to connector K_3.

The power-up reset pulse for the microprocessor is furnished by R_5-C_1. Pin P3.2 ($\overline{INT0}$) has a special function in this circuit because it acts as the Command Mode pin. If it is low immediately after reset, the MatchBox computer starts in command mode, and is able to receive new programs via the serial link to the PC. If this pin is logic high after a reset, the MatchBox computer starts to execute the user program stored in EEPROM IC_3.

All other CPU pins are available on connectors for user applications. Quartz crystal X_1 determines the clock speed of the processor. It also sets the baud rate (data speed) on the serial link between the MBC and the PC. Diode D_1 acts as a polarity reversal protection, while regulator IC_4 allows

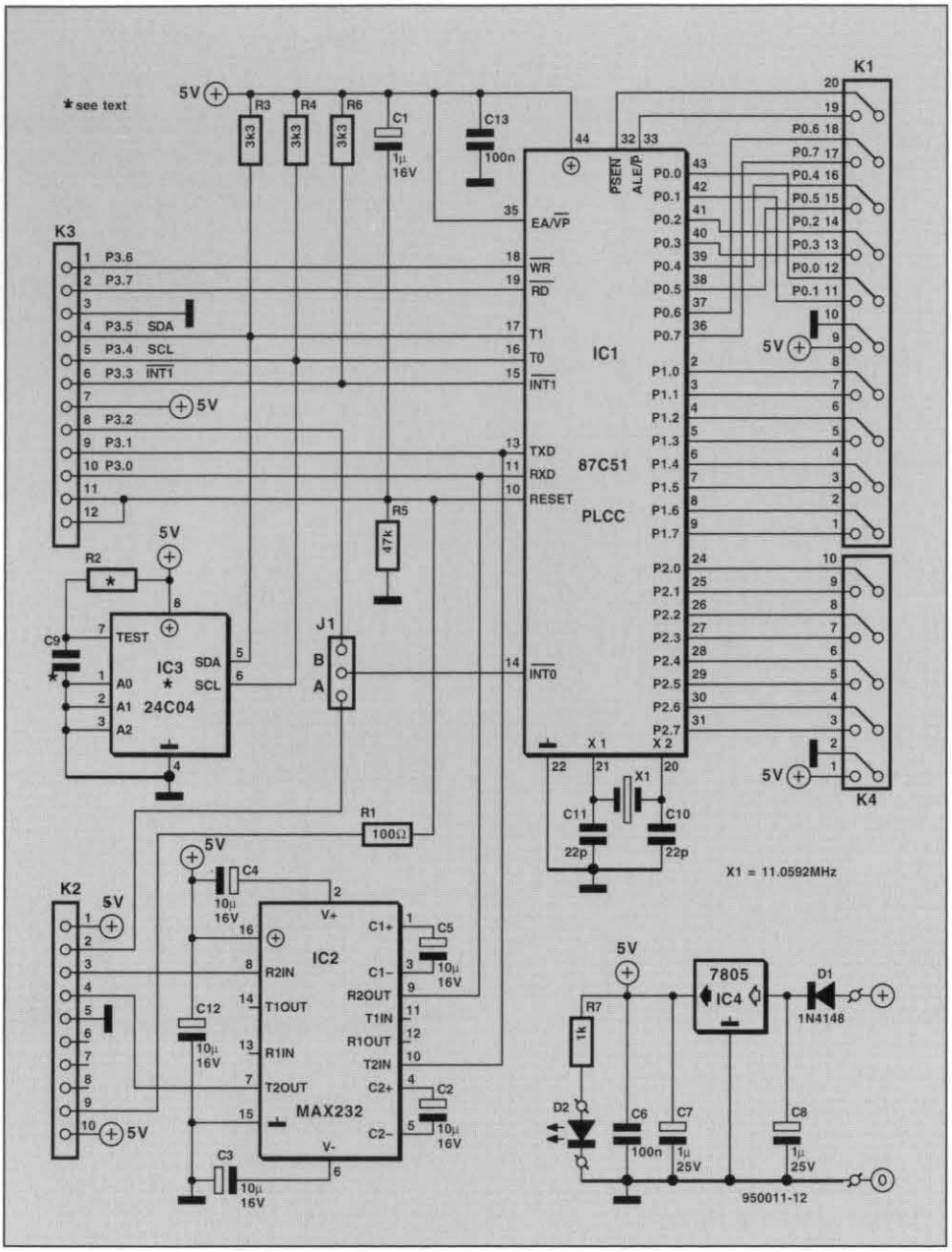


Fig. 2. Circuit diagram of the MatchBox computer.

EEPROM	R2	C9	Capacity
PCF8582A	56 kΩ	3.3nF	256 bytes
PCF8582B	n.f.	n.f.	256 bytes
PCF8582C	n.f.	n.f.	256 bytes
24C04B1	n.f.	link	512 bytes
22C08B1	n.f.	link	1024 bytes
X24C16	n.f.	link	2048 bytes

Table 1. EEPROM options. n.f. = not fitted.

the MBC to be powered by a 9 to 15 V d.c. supply.

Options

Depending on your requirements and applications, and the different types of EEPROM available, the computer offers a number of configurations. The available options are discussed briefly below.

Reset options

If you want to connect an external watchdog or reset controller, capacitor C_1 must be omitted. If necessary, the reset presskey may also have to be connected in a different way.

EEPROM options

The circuit allows different types of EEPROM to be used. Depending on the type you want to use, capacitor C_9 and resistor R_2 have to be fitted, omitted or replaced by a wire link. The configurations are listed in **Table 1**. The designer has successfully tested the types PCF8583C, ST24C04B1, ST24C08B1 and X24C164. Although he other EEPROMs listed have not been tested, they should also work.

Supply options

The on-board voltage regulator, IC_4, allows you to use almost any small d.c. supply with an output voltage between 9 V and 15 V to power the MBC. If you have a 5-V supply, IC_4 is omitted and replaced by a wire link.

Quartz crystal options

Although you may, in principle, use any quartz crystal with a frequency between 4 MHz and 12 MHz, it should be noted that the quartz frequency determines not only the CPU speed, but also the baud rate of the serial link via which the downloads are made. Choosing, say, 8 MHz, may work as far as the user programs are concerned, but it will result in a baud rate which is not supported by your PC. The recommended quartz frequency is 11.0592 MHz. This gives standard baud rates while being near the maximum speed of the processor. Although you may find that the CPU works up to 16 MHz, downloads to the EEPROM will not be reliable at clock speeds in excess of 12 MHz. Furthermore, all timing values used in the example programs on the disk with this project are based on a clock of 11.0592 MHz. The use of a different quartz frequency is not recommended unless you have some experience.

How it works

After a reset, the 87C51 starts to execute the code contained in its internal ROM. This code, as already mentioned, forms the 'invisible' operating system of the MBC. One of the first actions is a check on the level of the $\overline{INT0}$ line (P3.2). If a low level is read, the system enters the Command Mode. In this mode, the user program may be loaded into EEPROM IC_3, and started. This is normally done with the aid of the MatchBox compiler, in particular, with the integrated Terminal Emulator. If the processor reads pin P3.2 as high, the operating system starts to execute the user program stored in the EEPROM. The EEPROM also tells the operating system whether or not further EEPROMs are connected which also contain parts of the user program. This information is obtained by fetching the configuration parameters from the EEPROM before the start of the user program. Because of this, the socket for IC_3 should always contain an EEPROM having the I^2C address 10100000_B, irrespective of whether any further EEPROMs are connected.

Next, individual bytes are fetched from the EEPROM, and interpreted. As already mentioned, the EEPROM does not contain 8051 machine code, but a special type of code. Because each byte is addressed and fetched individually,

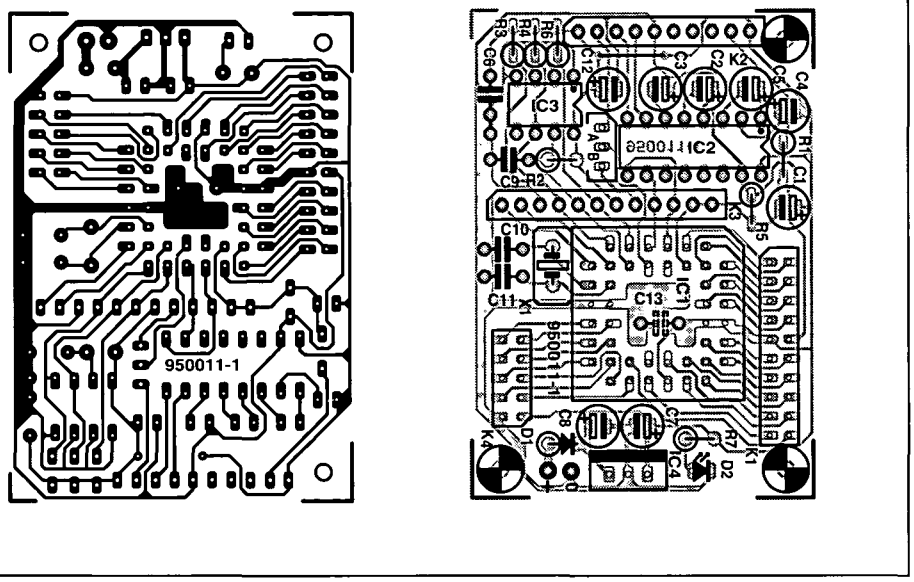


Fig. 3. Artwork for the PCB designed for the MatchBox computer (board available ready-made).

COMPONENTS LIST

Resistors:
R_1 = 100Ω
R_2 = see text
R_3,R_4;R_6 = 3kΩ3
R_5 = 47kΩ
R_7 = 1kΩ

Capacitors:
C_1 = 1µF 16V radial
C_2-C_5,C_{12} = 10µF 16V radial
C_6,C_{13} = 100nF
C_7,C_8 = 1µF 25V
C_9 = see text
C_{10},C_{11} = 22pF, raster 5mm

Semiconductors:
D_1 = 1N4148
D_2 = LED, 3mm
IC_1 = 87C51 (PLCC44) (order code 956508-1)
IC_2 = MAX232
IC_3 = 24C04 (see text)
IC_4 = 7805

Miscellaneous:
K_1 = 20-way boxheader.
K_2 = 10-way SIL pinheader.
K_3 = 12-way SIL pinheader.
K_4 = 10-way boxheader.
X_1 = quarz crystal 11.0592MHz.
J1 = 3-way pinheader plus jumper.
PLCC44 socket.
8-pin DIL IC socket.
16-pin DIL IC socket.
PCB and programmed 87C51 (set); order code 950011-1, see page 70.
Examples/utilities diskette: order code 956009-1, see page 70.

the execution speed of MatchBox programs is not very high. For many practical applications, however, the speed is more than sufficient. If you want to implement high speed program parts, there is always the possibility of calling 8051 assembler routines.

The selection between 'stand alone' or 'command' mode is determined by the level of the P3.2 line at start up. To make the selection as easy as possible, a jumper is provided. This is normally fitted in position 'A', when the user determines, via connector K_2, whether or not a download is desired. This requires P3.2 to be switched to ground. So, if you link pins 2 and 5 of the socket pushed onto pinheader K_2, the MatchBox goes into Command Mode after a reset. If you remove the socket from the pinheader, the MatchBox is in Stand-alone Mode. In this way, a suitable download cable can help to simplify the practical use of the computer. If you want to use P3.2 yourself in Stand-alone Mode, JP_1 may be set to position 'B' , and pin P3.2 may be connected via pin 8 on pinheader K_3.

Construction and test

It is strongly recommended to build the computer using the ready-made printed circuit board supplied through our Readers services, or through a kit supplier. The layout of this board is given in **Fig. 3**. IC sockets should be used for all ICs. In particular with the socket for IC_1, great care should be taken to fit it the right way around on the PCB. Before soldering, check the position of the bevelled edge of the PLCC socket against the mark on the

K1
20-way boxheader

1	P1.7
2	P1.6
3	P1.5
4	P1.4
5	P1.3
6	P1.2
7	P1.1
8	P1.0
9	+5 volt
10	Ground
11	P0.1
12	P0.0
13	P0.3
14	P0.2
15	P0.5
16	P0.4
17	P0.7
18	P0.6
19	ALE
20	$\overline{PSEN}$

K2
10-way, single row pinheader, remove pin 8

1	Vcc +5 V
2	Command Mode actuate
3	RxD, serial data to MatchBox; to pin 3 (2) of 9 (25) way sub-D conn. on PC
4	TxD, serial data sent by MatchBox, to pin 2 (3) of 9 (25) way sub-D conn. on PC
5	Ground, to pin 5 (7) of 9 (25) way sub-D conn. on PC
6	n.c.
7	n.c.
8	n.c.; pin removed, socket polarizer
9	Reset switch
10	Reset switch (Vcc +5 V)

K3
12-way single row pinheader, remove pin 11

1	P3.6 $\overline{RD}$
2	P3.7 $\overline{WR}$
3	Ground
4	P3.5 I^2C SDA
5	P3.4 I^2C SCL
6	P3.3 I^2C INT1
7	Vcc +5 V
8	P3.2 INT0 if JP at pos. B
9	P3.1 TxD
10	P3.0 RxD
11	Reset, pin removed, socket polarizer
12	Reset

K4
10-way boxheader

1	+ 5 volt
2	Ground
3	P2.7
4	P2.6
5	P2.5
6	P2.4
7	P2.3
8	P2.2
9	P2.1
10	P2.0

Table 2. Connector pin functions.

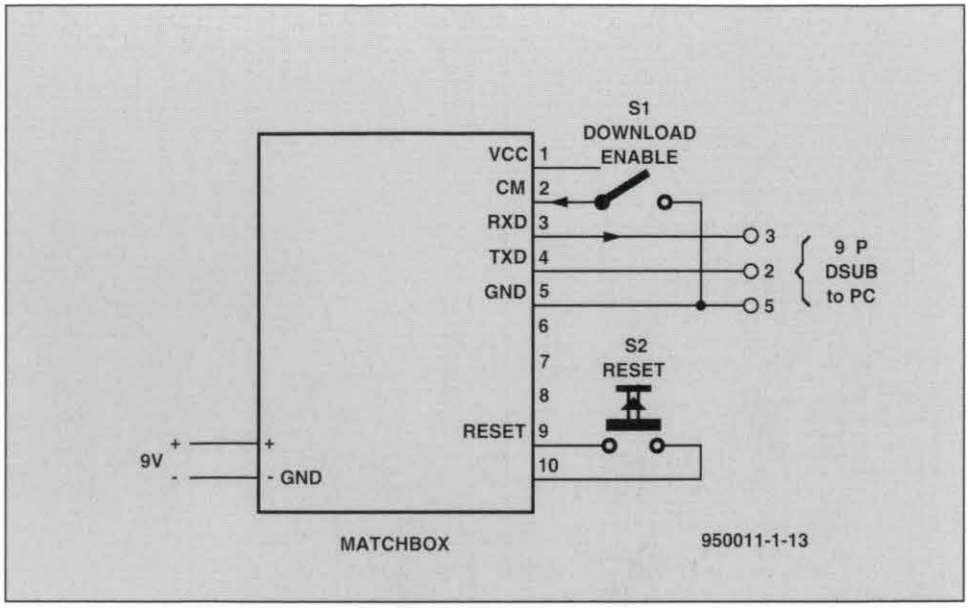


Fig. 4. Connection, via K2, of the MatchBox computer to the PC for an initial test.

component overlay. Capacitor C_{13} is fitted at the solder side of the board, beneath the processor. Do not forget the wire link! All capacitors are miniature types for upright mounting (radial).

Do not insert the ICs until you have checked the presence of the 5-V supply voltage at the relevant pins of the IC sockets. If this is okay, switch off, and insert the ICs, observing their orientation.

To prevent wrong connections on pinheaders K_2 and K_3, the pins mentioned in **Table 2** may be pulled out and inserted into the mating SIL socket. In this way, it is not possible to fit the socket the wrong way around. This method is especially recommended when supply voltages are carried via the connectors on the board.

The current consumption of the MatchBox BASIC computer is between 11 mA (without the MAX232 and the LED) at 5 V, and about 30 mA at 12 V with the MAX232 and the LED fitted.

After a last, thorough, visual inspection of the board, the computer is ready to be connected to the supply and the PC as shown in **Fig. 4**. Fit the jumper in position 'A' (download enable), and be sure to close switch S_1. On the PC, start the terminal utility (within the compiler), or run another communication program set to 19,200 bits/s (no handshaking). Switch on the MatchBox computer. The start up message

```
Matchbox V03 (TOP=66)
```

should appear on the PC screen. If not, press and release the reset key. The number behind the V indicates the version of the software. The TOP number is hexadecimal, and indicates to what extent the internal RAM is used. The number '66' indicates that internal RAM is available for user programs, starting at address 66H. Next, you can start programming, on which subject you will find more in the next instalments of this article. To whet your appetite, we already print an example file in **Fig. 6**.

Extensions

Because of the multitude of free port lines available on the MatchBox computer, connecting up extensions like LEDs, switches and keyboards (via suitable drivers) should not present problems. Example programs which help you with the software aspect of such extensions are available on the previously mentioned disk, and will be discussed in the following instalments. Likewise, the two communication channels of the MatchBox computer, RS232 and I^2C, also lend themselves to extensions. The I^2C bus in particular enables a host of extensions to be hooked up and programmed quite easily. Just look at the examples below.

I^2C RAM and EEPROM extension

The schematic in **Fig. 5** shows how additional EEPROMs or RAMs may be connected to increase the memory capacity of the computer. The I^2C bus addresses

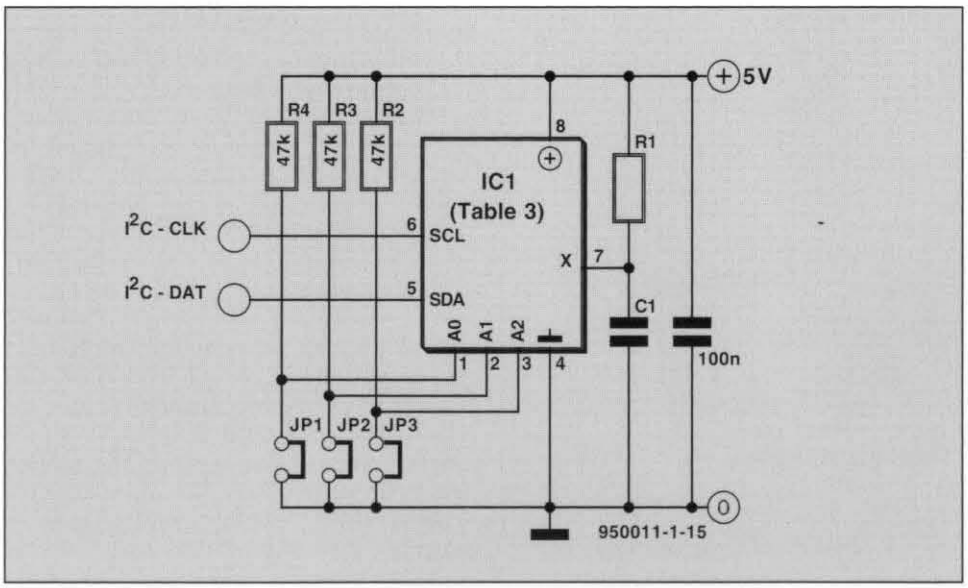


Fig. 5. Basic connection schematic for external I^2C RAMs and EEPROMs.

Device	R1	C1	capacity	memory
PCF8582A	56 kΩ	3.3 nF	256 bytes	EEPROM
PCF8582B	n.f.	n.f.	256 bytes	EEPROM
PCF8582C	n.f.	n.f.	256 bytes	EEPROM
24C04B1	n.f.	link	512 bytes	EEPROM
24C08B1	n.f.	link	1024 bytes	EEPROM
X24C16	n.f.	link	2048 bytes	EEPROM
PCF8570	n.f.	link	256 bytes	RAM
PCF8571	n.f.	link	256 bytes	RAM

Table 3. Memory extensions. n.f. = not fitted.

```
; MatchBox Running Light
;
RESOURCE I2C-EEPROM    256 BYTES @05000H ; program storage in EEPROM
RESOURCE 8051-RAM       10 BYTES @70H    ; data storage in 8051 RAM

INTEGER X               ;

  X:=00000001B          ; first LED on, all others off
LOOP:                   ; start of endless loop
 DELAY(100)             ; idle time
 P1:=NOT X              ; port output, LEDs are active low
 X:=X SHL 1             ; shift bit pattern
 IF X=100000000B THEN   ; test if shifted 8 times
   X:=1                 ; if yes then turn on first LED again
   ENDIF                ;
 GOTO LOOP              ; start over again
END
```

Fig. 6. Example of a MatchBox BASIC program (not on disk).

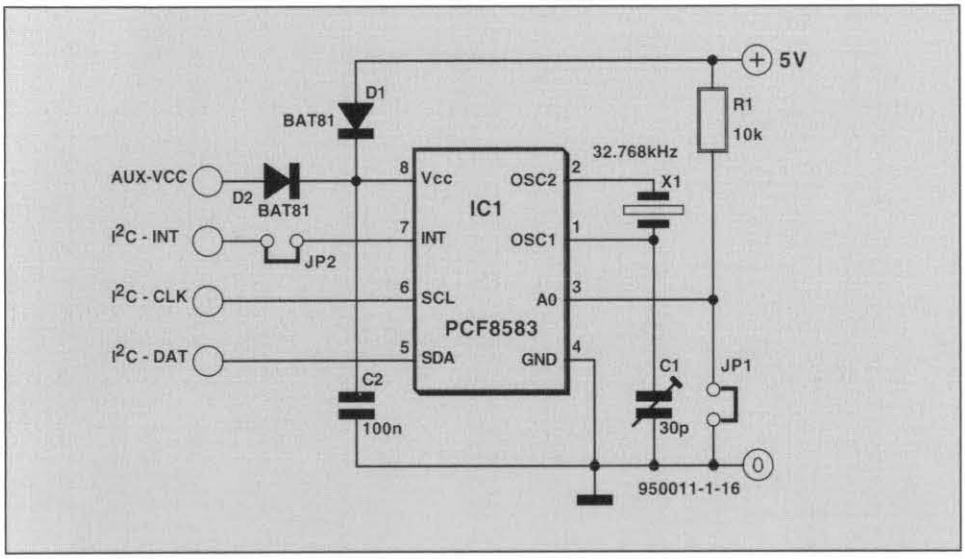


Fig. 7. Connecting up a real-time clock IC.

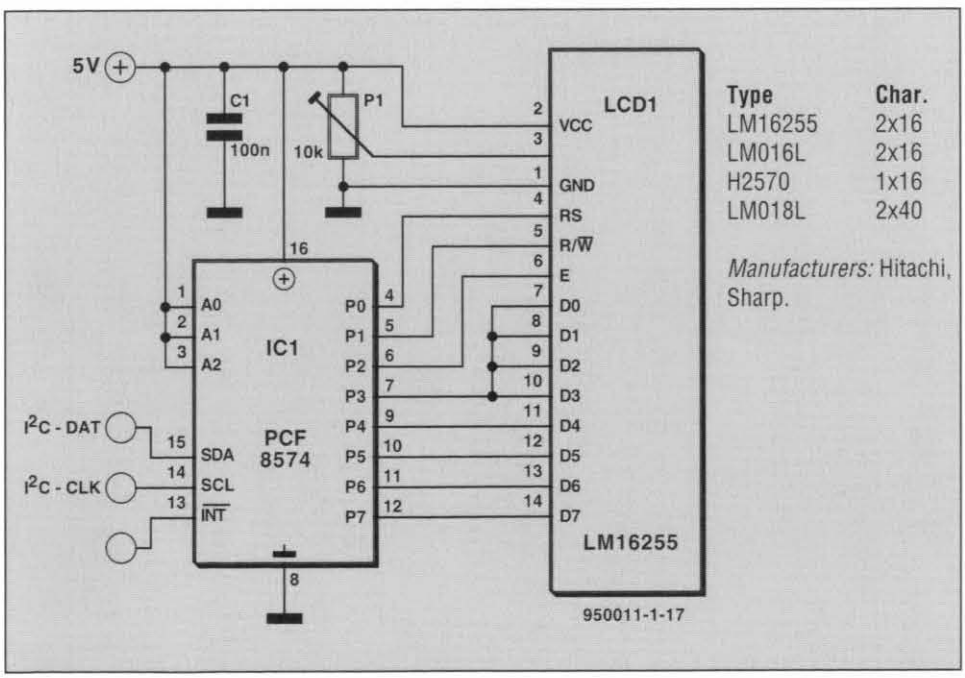


Fig. 8. Connecting up an LCD.

of the memory devices may be set via jumpers JP_1, JP_2 and JP_3. Depending on the type of EEPROM or RAM used, R_1 and/or C_1 may have to be fitted, omitted or replaced by a wire link. When connecting several extension, be sure to avoid address conflicts. The MatchBox compiler is informed of devices available for program storage by means of special declarations at the start of the BASIC program. More about this in the next instalments.

I²C real-time clock extension

Figure 7 shows how to connect a real-time clock IC to the MatchBox computer via the I²C bus. A (Lithium) battery may be connected to diode D_2 to make sure that the clock keeps ticking when the supply for the computer is switched off. the clock is calibrated with trimmer capacitor C_1. Jumper JP_1 sets the I²C address of the clock. When JP_2 is fitted, the clock is allowed to generate interrupts. By means of special declarations, the RAM contained in the PCF8583 can be made accessible for use by the MatchBox computer.

LCD connection via I²C

The circuit diagram in **Fig. 8** shows how the matchBox computer may be used to drive an LCD module with the aid of an I²C bus expander type PCF8574. The I²C bus address is fixed in this IC, which makes for simple and elegant control of the LCD in the MatchBox programming language. Because the relevant driver is already available, outputting texts and numbers (for instance, measurement values) to the LCD does not present problems. The inset in Fig. 8 lists the types of LCD which may be used in this application. The pinning shown in the circuit diagram refers to the LM16255 only. For the other types, consult their datasheets.

Other I²C extensions

The MatchBox computer may also be connected to any of the I²C compatible modules described in this magazine over the past few years (use the EIT to look them up). The computer has a built-in, universal, I²C driver to enable the communication with these modules. For instance, the computer is capable of driving a 7-segment display module, or an equalizer.

8051 external memory extension

Just like any other 8051-based processor system, the MatchBox may be connected to external RAM/EPROM memory via an address latch. To a certain extent, such an extension may also be used, for instance, to store data or 8051 assembler routines. This option is, however, for 8051 experts only, and should not be attempted by beginners. (950011-1)

PICTURE-IN-PICTURE PROCESSOR (Part 1)

Opinion is divided on the subject. Some say it's extremely useful, while others hold that it is only for restless TV addicts. Picture-in-picture (PIP) is a feature found on only the most expensive of television sets. The set-top PIP processor described here adds double viewing to any TV set having a SCART socket with video and RGB inputs. The second video source can be a video tape recorder, a satellite TV tuner, a camera, or even a video disk player.

Design by W. Sevenheck

THE fans claim that PIP is not just high-tech fancy stuff, but an extremely useful extension of the TV set's features. For example, by using PIP, you never get back too late to a film which was interrupted for a commercial break. That is, assuming that you started zapping across the other channels when the film was suddenly interrupted for the latest on washing powder or toothpaste. When one of the other channel happens to be interesting too, you may well forget about your film after a few minutes. With the PIP processor that can not happen because the small inset picture tells you exactly when the commercials end and the channel is safe to return to. Switch from inset to large screen, and you are back with your film again.

The use of the inset picture is not restricted to TV signals only. It is possible, for instance, to 'PIP' a camera signal for surveyance applications, for instance, to watch the children's bedrooms, the driveway, etc.

The external PIP processor described here requires two video signal sources. That is not an unusual requirement, because top-of-the range TV sets also have two built-in TV tuner sections to enable their PIP system to work. Fortunately, many owners of a TV without a PIP already have their second TV tuner: the one inside the video cassette recorder (VCR). The PIP processor described here is simply inserted in the SCART link between the VCR and the TV set. It offers all features offered by a very sophisticated and highly integrated PIP processor IC. In contrast with high-end TV sets offering PIP, the inset picture may be positioned at any location on the screen, while the size, brightness, colour settings and frame are also freely programmable to suit your preferences. The only condition is that your TV set has a SCART socket with RGB and Fast Blanking inputs.

Block diagram

The block diagram in **Fig. 1** shows that combining two video sources is not at all simple. The main problem is to keep the two pictures synchronized to each other. This function is performed by the 'heart' of the circuit, the integrated PIP processor type SDA9188-3x and the associated A-D converter type SDA9187-2X (both from Siemens Components). Apart from the synchronization, the PIP IC core also ensures that the 'small' picture is inset at the proper position in the large (main) picture.

The core of the circuit requires a composite video (CVBS) signal supplied by the PIP, as well as synchronization signals supplied by the two video sources. These functions are carried out by a number of 'smaller' ICs: a PAL decoder type TDA4510, a baseband delay circuit type TDA4661, and

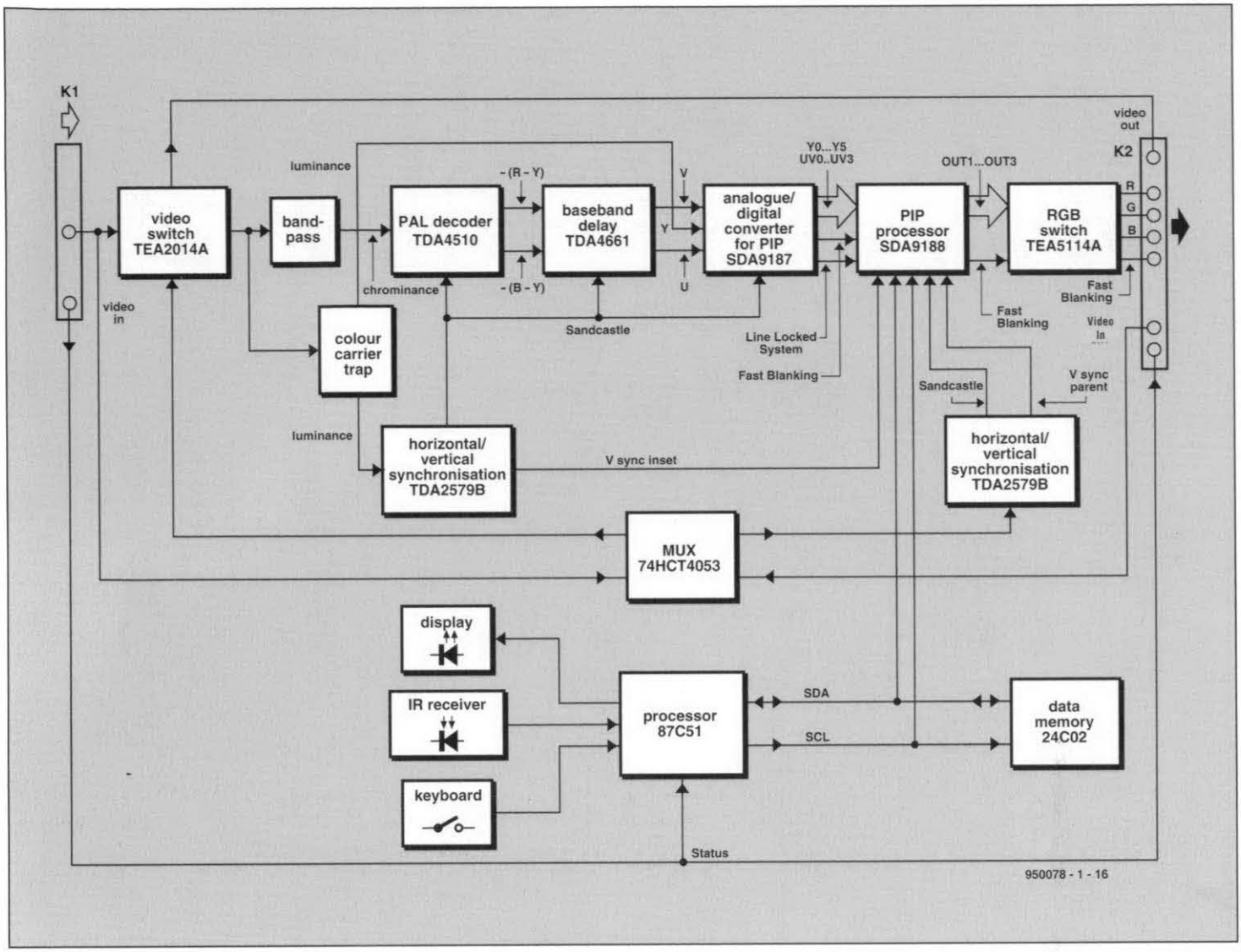


Fig. 1. Block diagram of the PIP processor. The core is formed by the Siemens SDA9188-3X which compresses the inset picture, and enables it to appear synchronized in the 'parent' picture.

two horizontal/vertical synchronization separators type TDA2579B. The RGB output signals arrive at the RGB pins of the SCART connector on the TV set via a type TEA5114A RGB switching circuit. The status information supplied by the PIP is also used as a switching signal.

A microcontroller type 87C51 translates commands received from a 'local' keyboard or an infrared remote control into I^2C format, and also determines the mode of operation of the PIP core. Data is stored in a non-volatile program memory formed by an EEPROM type 24C02. The status signal generated by the PIP processor also controls an analogue multiplexer (MUX, 74HCT4053), which determines the parent picture/inset picture assignment of the two video sources. The parent picture is the main, large, picture. The inset picture is the small picture (sometimes called PIP by convenience). In video recorder playback mode, the TV picture becomes the inset picture, and the VCR picture, the parent picture. All settings can be checked with the aid of ten LEDs.

The video signal paths

The full circuit diagram of the PIP processor is given in **Fig. 2**. While the audio pins of the SCART sockets are simply interconnected, the video signals are routed to the PIP processor via pin 20 of K_1 (video recorder) and pin 20 of K_2 (TV set), and first arrive at the video multiplexer (MUX). The video switch type TDA2014A is only used as a buffer/splitter. The analogue multiplexer type 74HCT4053 switches either the VCR signal or the TV signal to the input buffer, pin 3. This signal is buffered and re-appears at pin 2. The VCR signal, on the other hand, is amplified by a factor of 2 (independent of the position of the MUX), and fed to the video input of the TV set (pin 19 of SCART socket K_2). The video signal at the output (pin 2 of IC_1) is displayed as the inset picture. It is routed in various directions: to a colour carrier trap, R_{10}-C_{15}-L_4; to one of the sync separators, IC_6; and to a voltage divider, R_{12}-R_{13}, followed by a filter, L_3-C_{12}. The latter serves to extract the chrominance (colour) information and the colour burst from the composite video signal.

The chrominance signal arrives at a type TDA4510 PAL decoder IC. Traditionally, this IC is accompanied by a glass-based delay line to store the chrominance information of the previous line, allowing a special circuit to add the information contained in two successive lines, and so correct any colour errors. In the PAL system, the delay line delays the (R–Y) and –(B–Y) components modulated onto the 4.43-MHz subcarrier, for adding to the non-delayed signal. This addition is critical in respect of phase position and level.

Here, the TDA4510 uses an electronic equivalent of the 'old' glass delay line, a baseband delay circuit type TDA4661. The two colour difference signal outputs, pins 1 and 4 of IC_4, supply demodulated difference

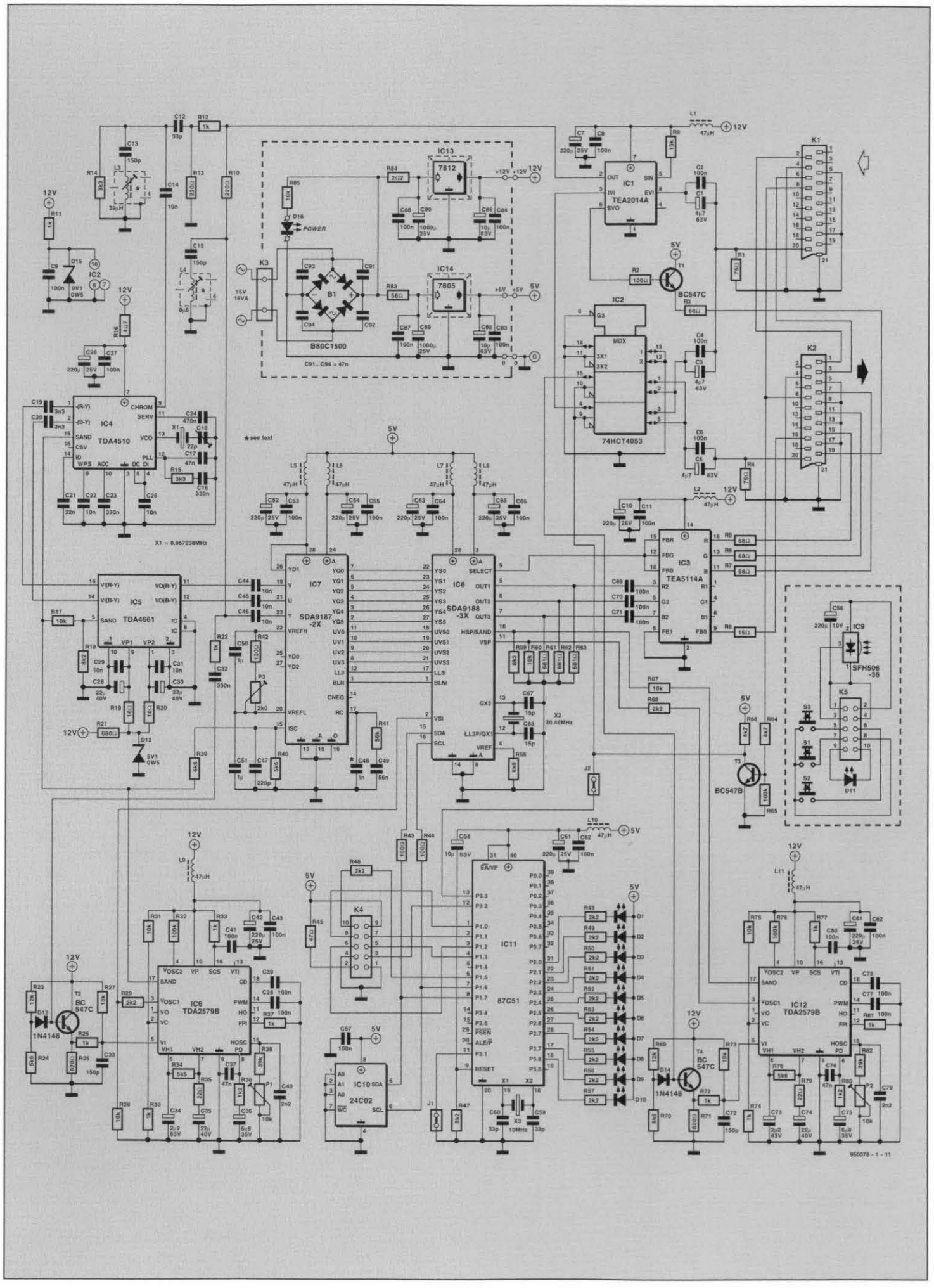


Fig. 2. Circuit diagram of the PIP processor.

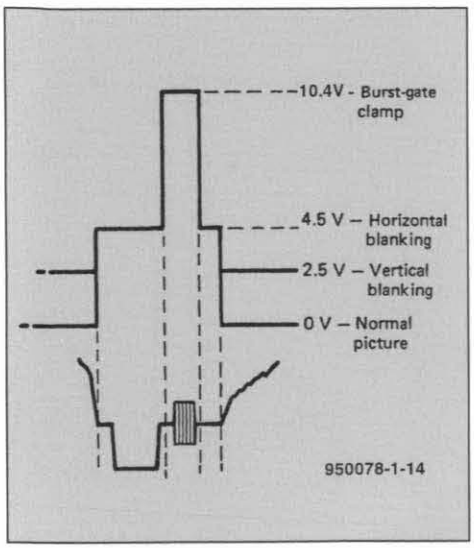


Fig. 3. The sandcastle pulse contains timing information about the colour burst, H/V sync pulses and the picture itself.

signals (hence 'baseband') to IC_5, which copies them into an analogue shift register. After propagating through the shift register, the 64-µs delayed signal and the non-delayed signal are added in the IC to give the traditional –(R–Y) and –(B–Y) components. The shift register is clocked by a PLL circuit contained in the TDA4661. The line frequency extracted from the 'sandcastle' pulse is used as the reference frequency for the PLL.

The video signal is also applied to pin 5 of IC_6. The TDA2579B has many applications, of which only two are used here. The main function of IC_6 is to generate the so-called sandcastle pulse (**Fig. 3**), whose position is determined by the centre of the horizontal synchronization pulse contained in the video input signal. The sandcastle pulse is a three-level waveform which supplies timing references on four states:

0 V:	normal picture reference level
2.5 V:	vertical synchronization
4.5 V:	horizontal blanking
10.4 V:	burst-gate clamp

The sandcastle pulse supplied by IC_6 synchronizes the PAL propagation time demodulator (i.e., PAL decoder plus delay line), as well as the A-D converter for the inset picture. Pin 3 of IC_6 supplies narrow pulses which depend on the vertical synchronization pulse. These pulses are also fed to the PIP processor.

The second TDA2579B (IC_{12}) supplies the same information (sandcastle and vertical sync) derived from the parent video signal, and also drives the PIP core.

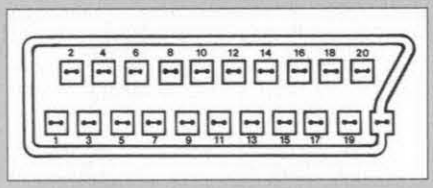
The PIP-plus chip set

Briefly recapitulating, the following signals are available at the inputs of the PIP chip set:

- luminance (brightness) of the inset;
- colour difference of the inset;
- sandcastle signal of the inset;
- vertical sync pulses of the inset;
- sandcastle signal of the parent;
- vertical sync of the parent.

The analogue signals (luminance and chrominance) need to be digitized before they can be processed by the PIP processor. The A-to-D conversion is the main function of the SDA9187, which was developed specially for that application. As indicated by the block diagram of this IC (**Fig. 4**), the luminance signal (YIN) and colour difference signals (UIN and VIN) are clamped and then applied to three analogue-to-digital converters with a resolution of 6 bits. The clamping levels (i.e., the drive margin of the signals) are determined by the voltage difference between VREFH and VREFL (black level). The luminance information is available as a six-bit wide word representing a value between 0 and

SCART SOCKET PIN FUNCTIONS



1	Output	Audio, right channel
2	Input	Audio, right channel
3	Output	Audio, left channel
4	Ground	Audio
5	Ground	Blue
6	Input	Audio, left channel
7	Input	Blue
8		Function switch
9	Ground	Green
10		
11	Input	Green
12	Input	V-Sync
13	Ground	Red
14	Ground	Fast blanking
15	Input	Red
16	Input	Fast Blanking (H-Sync)
17	Ground	CVBS output
18	Ground	CVBS input
19	Output	CVBS
20	Input	CVBS
21	Ground	Plug/socket screening

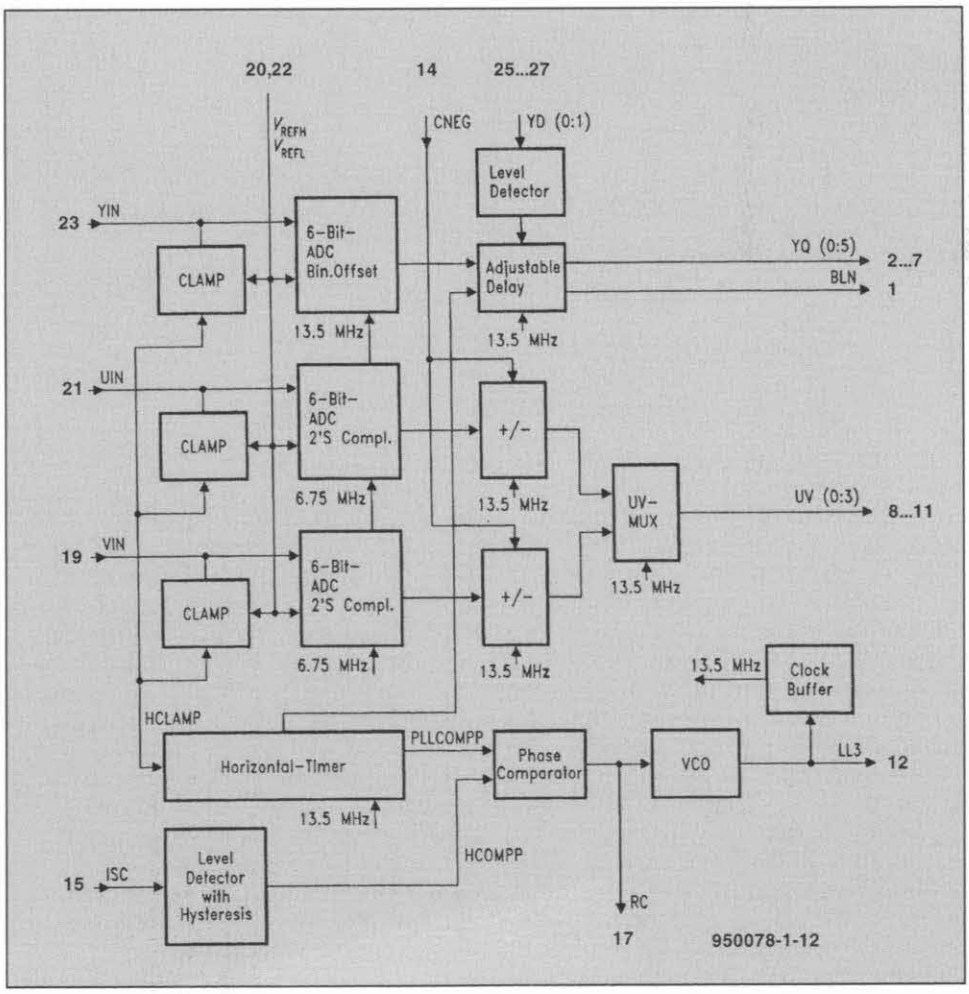


Fig. 4. Internal circuit of the SDA9187-2x, which serves, among others, to digitize the analogue luminance and chrominance signals (brightness and colour, respectively).

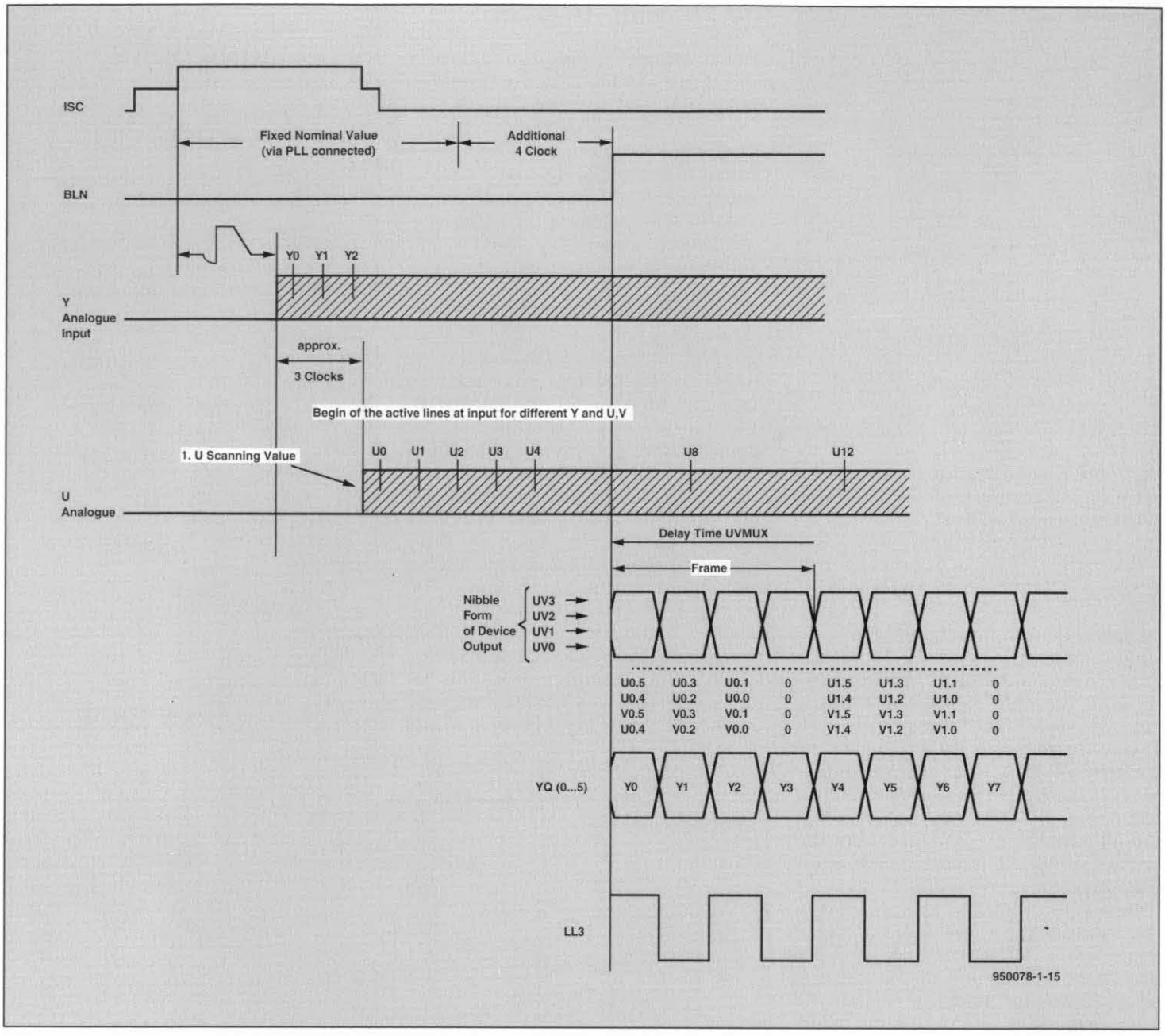


Fig. 5. Timing diagram for the various analogue and digital signals.

63. If necessary, this signal may be delayed to compensate propagation delay differences in the decoder that precedes the ADC. The levels of pins YD0, YD1 and YD2 here set a delay of 296 ns, which corresponds to four periods of the LL3 clock.

The SDA9187-2X has a PLL-based horizontal synchronization section consisting of a horizontal timer, a phase comparator and a VCO. The VCO generates a clock of 13.5 MHz which is phase locked to the video input signal. This clock is fed to the PIP IC (line-locked PIP system clock, LL3) as well as to the ADC, the delay line and the colour difference multiplexer. The horizonal scaler divides the LL3 signal by 864, and applies it to the phase comparator as a 'horizontal' reference. The external horizontal sync signal is derived from the sandcastle pulse with the aid of a level detector, and is blended in with the LL3 signal as regards the pulse width (345 LL3 periods). The frequency and phase sensitive digital phase comparator produces up/down current pulses which are turned into the VCO control voltage by smoothing in an RC network. The horizonal timer also determines the start and the width of the clamp pulses, as well as the position of the blanking pulses, BLN, which, in turn, determine the length of the brightness information. The BLN signal has to be given the same delay as the Y signal. When the blanking signal is high, the parent picture is synchronously replaced by the inset picture.

If desired, the colour difference signals may be inverted by using the CNEG input. This results in an inverted inset picture. This feature, although interesting, is not used here.

The digitized colour difference signals U and V (each having values between –32 and +31) arrive at a multiplexer which captures each fourth U and V sample, and copies it in half byte (nibble) format to outputs UV0 through UV3.

Figure 5 shows the complex relations between the information representing chrominance, luminance and horizontal/vertical synchronization at the inputs and outputs.

The components that make up the video signal of the inset source are applied to the PIP processor in digitized form. The resolution of the digitized input signals is six bits at a sampling rate of 13.5 MHz (luminance) or 3.375 MHz (chrominance). The processor combines the (till then non-synchronized) picture sources in such a

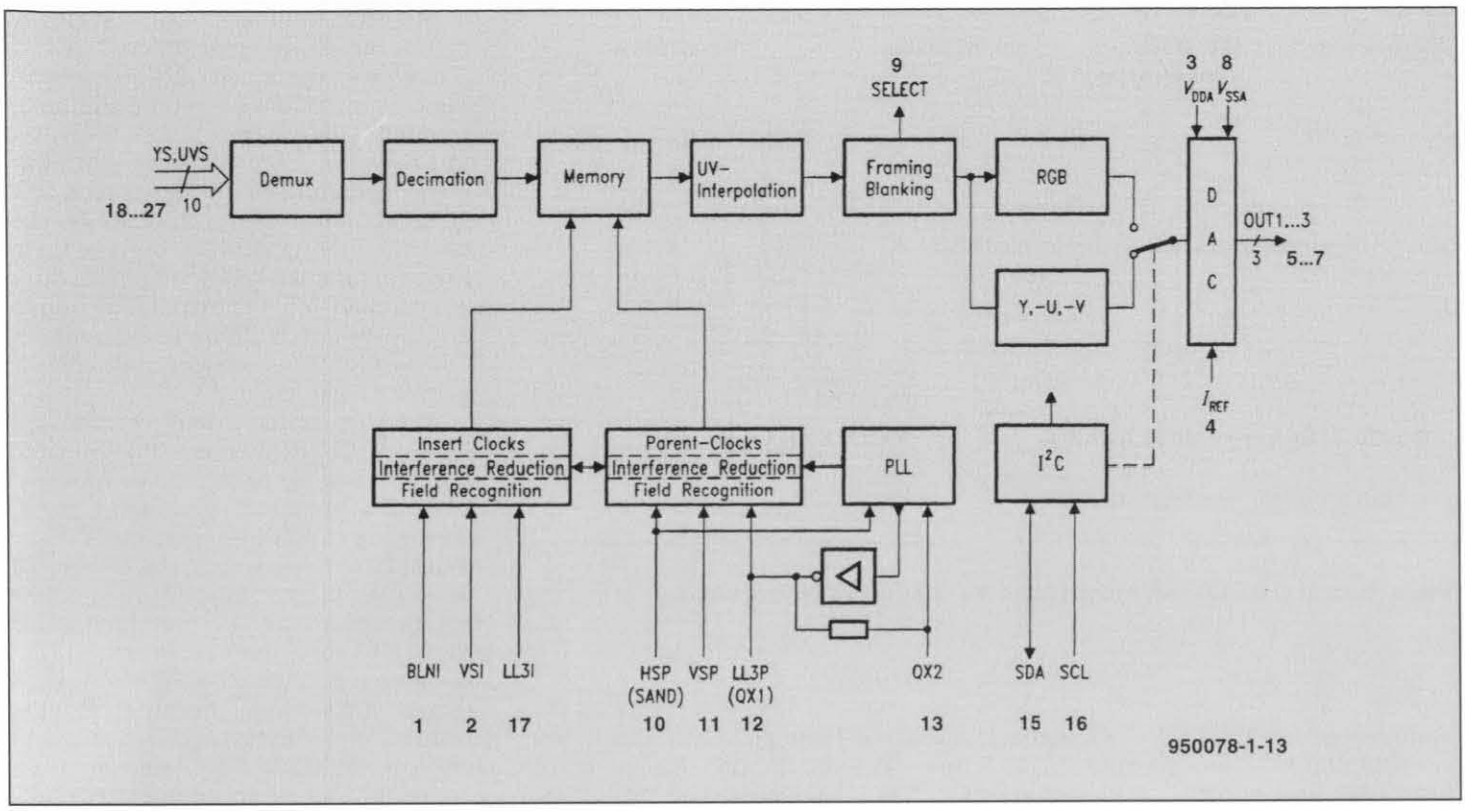


Fig. 6. Architecture of the heart of the PIP processor, the SDA9188-3X from Siemens.

way that a size-reduced inset is displayed in a corner of the 'normal' picture. Apart from the picture size reduction (based on compression using horizontally and vertically operating filters), the SDA9188-3X also stores the inset in an integrated picture memory with a total size of 89×212×9=169,812 bits. It also takes care of outputting this picture again at ⅓th or 1/16th of the original size.

The colour of the frame around the inset picture is programmable. The same goes for the position of the inset on the screen.

The output signals of the SDA9188-3X are available as analogue RGB or chrominance/luminance signals [(Y, –(B–Y), –(R–Y)]. The digital-to-analogue conversion is wideband for all components, and has a resolution of six bits.

The SDA9188-3X is capable of processing two picture formats: 50 Hz/625 lines and 60 Hz/525 lines. The ability of the chip to use 100-Hz scanning (or 120 Hz) allows it to be used in systems for standards conversion. A frame frequency of 50 Hz (or 60 Hz) may also be used, with automatic adjustment of the number of lines.

Problems may occur if the parent and inset pictures have different raster frequencies. With an unfavourable and unstable phase position of the sync signal and the parent clock, the inset picture may 'jump' by one line or one pixel, resulting in instability. Likewise, poor sync signals in the inset channel may cause instability too, though less serious.

The inset picture is synchronized to the parent picture with the aid of the horizontal and vertical sync signals SAND and VSP. The clock speed of LL3P equals 13.5 MHz, or 27 MHz when the system is used with 100-Hz scan (LL1.5P). These frequencies need to be generated by an internal PLL, because an external clock, LL3P, would only be usable in an all-digital environment (of the parent source). The

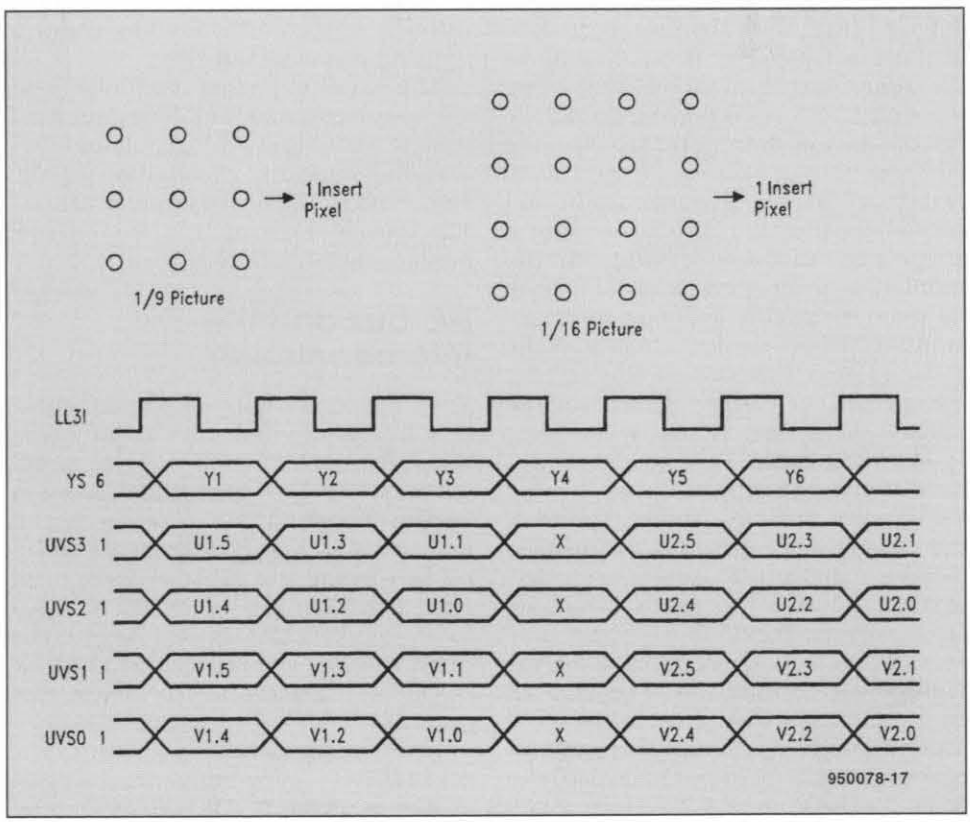


Fig. 7. Principle of 9-to-1 or 16-to-1 pixel compression technique.

Picture size	TV standard (lines/raster)	No. of pixels	No. of lines		
			Y	U	V
1/9	625	212	53	53	88
1/16	625	160	40	40	66

Table 1. Organization of the inset picture memory.

Start	Ack	Ack	Ack
S 0010 1110 A		0000 XXXX A	XXXX XXXX A P
Chip address		Register address	Data word Stop

Table 2. Format of the I²C commands sent to the SDA9188-3X PIP processor.

frequency of the internal clock signal is calculated as $f_T=864\times f_{SAND}$. Horizontal and vertical sync signals BLNI and VSI, as well as LL3I also serve to keep the inset picture synchronized to the parent picture. The interface between the inset and the parent picture is formed by an integrated picture memory, in which data is clocked in at the rate of the inset, and clocked out at the rate of the parent. Output signal SELECT (output signals of SDA9188-3x valid) controls an electronic switch in the TV set which frames the size-reduced inset into the parent picture.

Of particular interest in the block diagram (**Fig. 6**) is the data reduction of the inset picture. The data rate at the inputs YS4 through YS4 and UVS0 through UVS3 is 13.5 MHz. To reduce the amount of data to be stored, nine (PIP size ⅓th) or sixteen (PIP size ¼th) pixels are averaged horizontally and vertically, and then processed into a single pixel in the inset (**Fig. 7**). The number of pixels and lines that make up the inset picture depends on the TV standard of the inset source, and the picture size. The size of the picture memory is, of course, based on the maximum number of pixels and lines as shown in **Table 1**.

At the parent side of the IC, behind the picture memory, the data rate of the chrominance signal is quadrupled to match that of the luminance information. This is done to eliminate cross-colour effects after the D-A conversion, as well as to prepare for the digital RGB conversion. The rate-up conversion is achieved by linear (U-V) interpolation. A digital RGB matrix converts the Y, U and V levels read from the RAM into RGB data, which are then applied to three 6-bit D-A converters. Each of these DACs sends a current through a pull-down resistor, R_{61}, R_{62} and R_{63} (OUT1=R, OUT2=G, OUT3=B). The output current may be changed by modifying the resistors, or by programming the processor.

The output signals of the PIP processor then arrive at the TEA5114A, before they leave the circuit via SCART connector K_2. The TEA5114A only serves to amplify the RGB signals about 2 times, and to clamp them at about 2 V_{pp}. The BLANKING signal switches the RGB outputs off when there is no inset picture in the relevant picture lines. The BLANKING signal is connected through to the corresponding connections on K_2 via pins 9/10.

The power supply of the PIP processor is conventional, and based on fixed voltage +12 V and +5 V regulators. R_{83} and R_{84} limit the dissipation of the 7812 and 7805 to reasonable values. The 'power' LED on the front panel provides an on/off indication.

I²C bus and the microcontroller

The PIP IC core obtains all configuration information via its I²C interface. The PIP processor itself is found at address 00101110 = $2E_H$. When power is applied, an automatic reset is issued, and the SDA and SCL lines are freed. All bits in the ten registers, except bit PL27 (bit d3 in register 0) are then set to 0, while PL27 is set to 1. The SDA9188-3x is switched to 'slave' mode, and operates only when the inset clock, LL3I, is present.

I²C commands and data sent to the SDA9188-3X have the general format shown in **Table 2**. All registers increment automatically to the next one after a byte has been written.

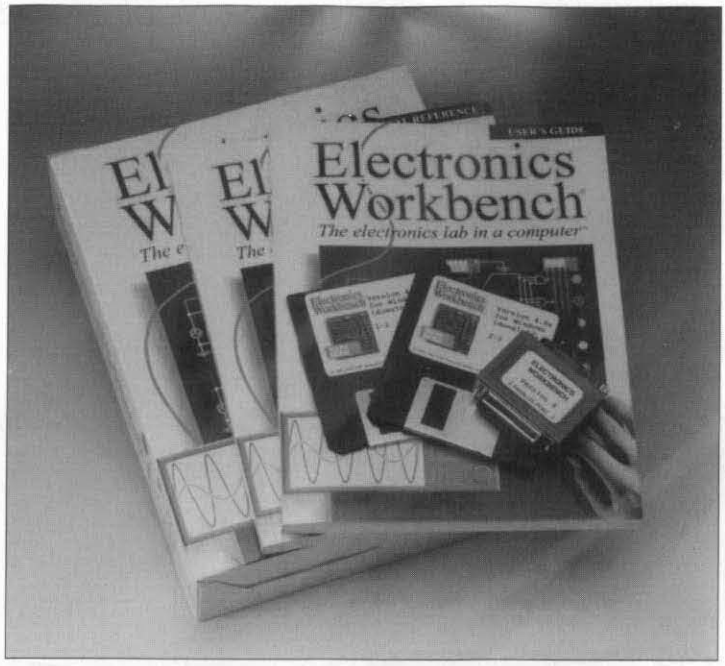
The vast number of options offered by the registers in the PIP core call for a microcontroller. The 87C51 is operated in an 'absolute minimum' environment here. The power-up reset pulse for the complete processor hardware is furnished by R_{47}-C_{58}. A 10-MHz quartz crystal, X_3, clocks the controller at 10 MHz. To prevent them from vanishing forever when the power is switched off, the register contents are copied into a 256-byte I²C compatible EEPROM, which is a non-volatile memory component.

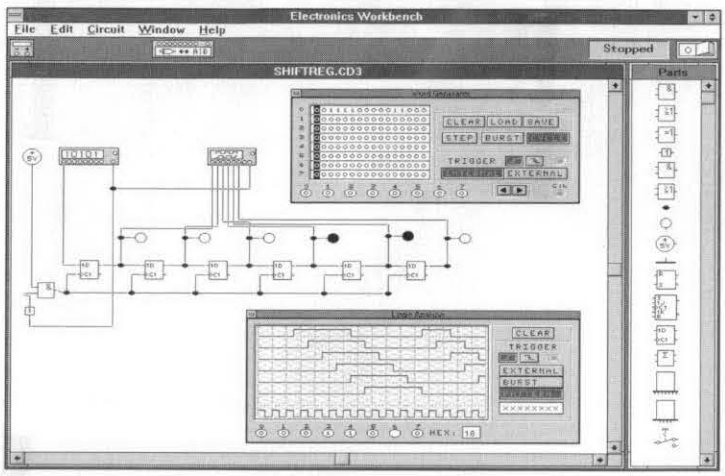
Register contents may be modified by using the RC5 compatible infrared remote control, or the three function keys on the PIP processor itself. Obviously the three presskeys have multiple functions to give access to all available options. Apart from generating the necessary I²C commands for the PIP IC core, the 87C51 also unravels and translates the RC5 codes received from the infrared remote control. The infrared receiver is based on the SFH506-36 detector from Siemens, and is accommodated on a separate printed circuit board together with the three presskeys that form the 'local' keyboard. This sub-board is connected to the main board vi a short length of flatcable. LED D_{11} lights when a valid (RC5 compatible) infrared signal is received. The other ten LEDs (D_1 through D_{10}) indicate the status of the microcontroller in programming mode. (950078-1)

The programming, construction and setting up of the PIP processor will be discussed in next month's second and final instalment.

Electronics Workbench v. 4.0

At a time when practically every electronics enthusiast has a PC at his or her disposal, it is not surprising to see more and more programs being introduced for the simulation of electronic circuits. A program called Electronics Workbench from Toronto-based Interactive Image Technologies Ltd. uses a less conventional approach to 'mimic' the behaviour of electronic components and circuits. It virtually simulates a complete electronics laboratory on the PC screen, offering to the designer a wide range of components, as well as his trusty measurement instruments.





Note: screendump produced with v. 3 of the program.

Recent surveys indicate that the majority of our readers have a computer, usually, a PC. However, not all electronics professionals and hobbyists use that wonderful computer as a tool for designing or testing circuits. That is surprising, because today's PC has sufficient computing power for that application. The problem, it seems, lies with the existing electronics simulation programs. Most of these are clearly aimed at the professional user, and that usually means a high price and a complex program structure and *ditto* operation. Although the operation has been improved significantly over the past few years, most Spice-like programs are still far from easy to use. By the way, those of you interested in this type of program may like to know that both PSpice and MicroCap are available as student or evaluation versions at reduced prices. The performance of these student versions is usually limited in respect of the number of nodes (component junctions) that may be used. Unfortunately, though the price is reduced, these programs are often still too complex for the hobby user.

A different approach

Electronics Workbench allows analogue and digital circuits to be simulated in a clear and educational way. The designers of the software have based the program on the measurement equipment an average electronics hobbyist may reasonably be expected to have available in the workshop. All these (virtual) instruments may be connected to a circuit which has been constructed (virtually, again) on the screen. Connecting and using the instruments is straightforward, and the whole operation looks as if one is actually measuring on a prototype circuit.

The recently released version 4.0 of Electronics Workbench has number of improvements over the previous version, 3.0. Fortunately, the people at Interactive Image Technologies do seem to listen to the users, as well as read reviews on their products. The most important, and very welcome, change from the earlier version is the full integration of the digital and analogue simulation. From now on, any circuit design may contain both analogue and digital sections, or a combination of these ('mixed-mode'). The separate programs for evaluation of analogue and digital circuits have disappeared — instead, everything is now simulated in a single environment. Also, a large number of less conspicuous changes have been made to the program. For instance, the oscilloscope has obtained a zoom function, and linear components such as the potentiometer, variable capacitor and variable self-inductance have been added. Also new is the support for hybrid components such as the A-D converter and its counterpart the D-A converter, the monostable multivibrator and the widely known 555. With the digital components we were pleased to see that support is now available for full adders, 3-to-8 segment decoders, 8-to-3 priority decoders, BCD-to-7-segment decoders and 1-of-8 multiplexers and demultiplexers.

Professional users will be interested in obtaining the Electronics Workbench Engineer's Pack. This version of the program offers three extra functions: an extra set of 2,000 component models, exporting and importing to and from Spice, and exporting to PCB design programs.

The basic version of Electronics Workbench 4.0 now consists of schematic capture, mixed-mode simulation and a library containing 350 component models (Model Set 1). This will set you back £199 (excl. VAT and P&P). The Engineer's Pack with 2,000 models (Model Pack 1 through 5) and the above mentioned import and export options costs £399 (excl. VAT and P&P).

Electronics Workbench is available in DOS or MS-Windows versions, as well as one for the Apple Macintosh.

(EA-1464)

For further information, contact Robinson Marshall (Europe) Plc, Nadella Building, Progress Close, Leofric Business Park, Coventry, Warwickshire CV3 2TF, England. Telephone: (01203) 233-216. Fax: (01203) 233-210. E-mail: rme@cityscape.co.uk

ACTIVE ANTENNA SPLITTER

A single-IC amplifier is described that makes it possible in a simple way to apply the TV antenna signal to two or more TV receivers in the same household without any attenuation.

Design by L. Lemmens

There are few households nowadays which possess only one TV receiver. Unfortunately, normally there is only one antenna (or in some cases, cable) outlet. It is, of course, possible to connect two or more TV receivers in parallel to this single outlet, but this gives a serious signal deterioration. This is because the outlet should be terminated into a 75 Ω load. The input of radio and TV receivers is, indeed, 75 Ω and the impedance of the coaxial cable connecting the two is also 75 Ω (see **Fig. 1**).

Mismatching

When two receivers are connected in parallel, the resulting impedance is only 37.5 Ω, and with three receivers, only 25 Ω. This causes a severe mismatch with the antenna outlet, resulting in little power being transferred to any of the receivers.

It is no remedy to use, say, 50 Ω cable in the belief that this will reduce the impedance. In fact, it will attenuate the signal, probably also cause ghosting (signal reflections in the cable), and is susceptible to interference leaking into the cable.

With radio reception, these effects are not nearly as bothersome as in TV reception, but distortion of the signal may still occur.

Remedy

If one of the impedances in a system is not 75 Ω, an impedance transformer may be used between the antenna cable and the receiver. For instance, a 300 Ω dipole antenna, frequently used for the reception of FM radio stations in the 87.5–108 MHz band, is often supplied with a 300–75 Ω transformer to enable the connection between antenna and receiver to be made in 75 Ω cable (always assuming, of course, that the receiver has a 75 Ω input).

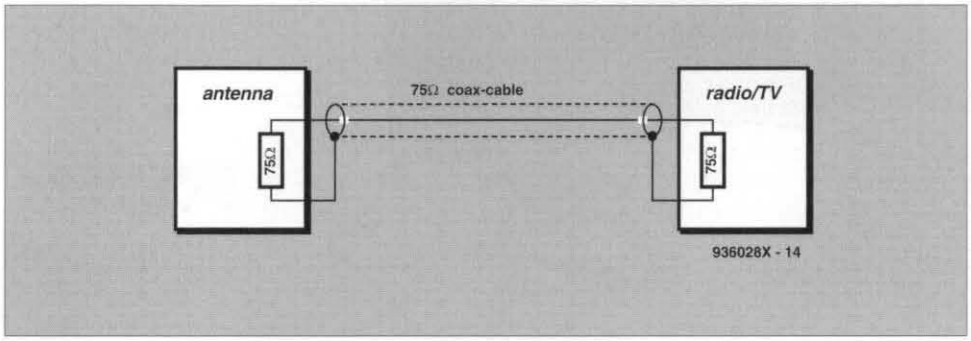


Fig. 1. Antenna, connecting cable and receiver must all have the same characteristic impedance: here, 75 Ω.

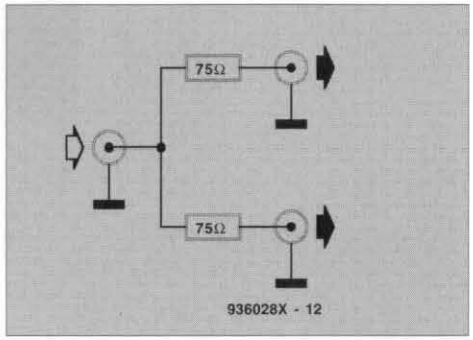


Fig. 2. This inexpensive splitter causes a serious mismatch.

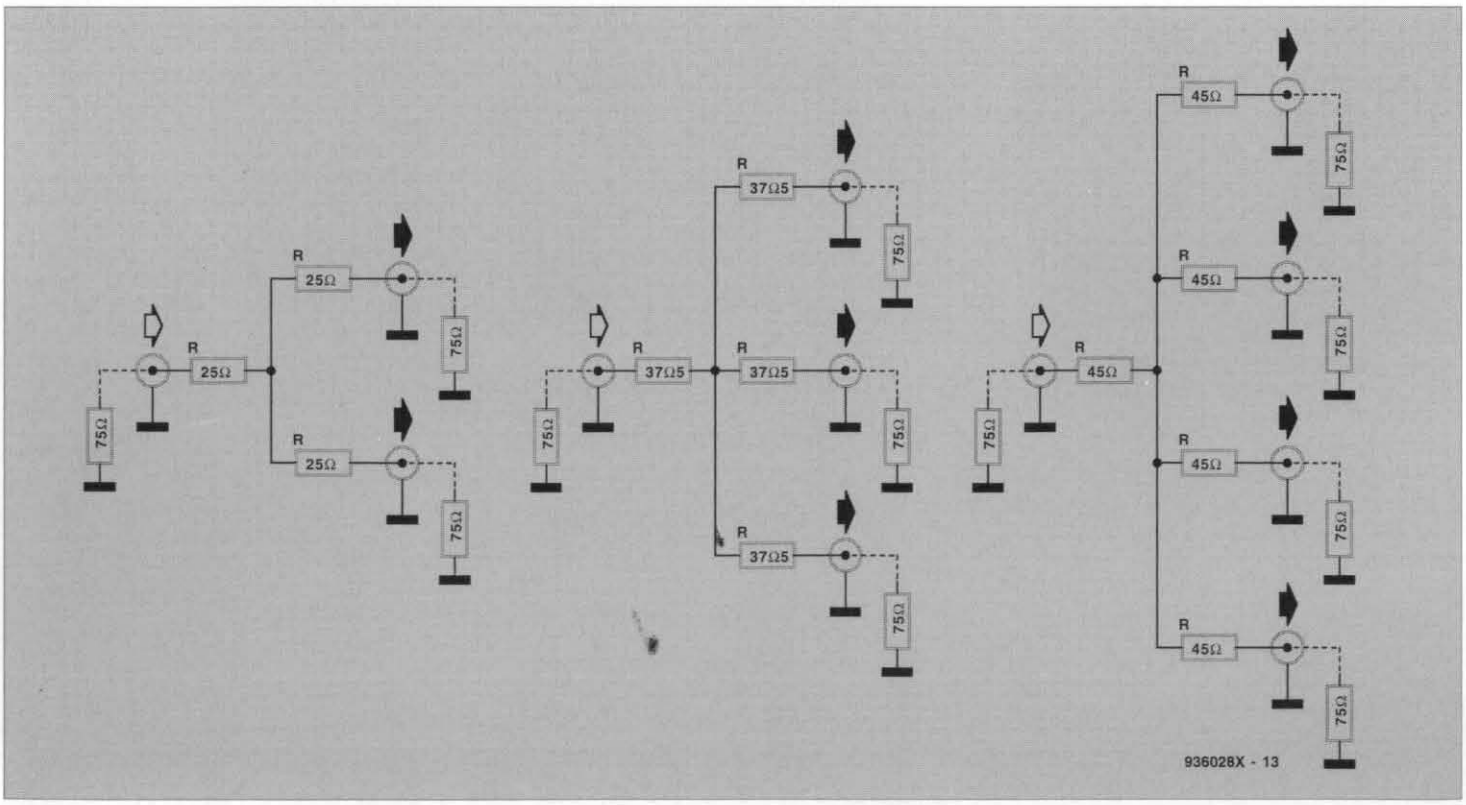


Fig. 3. These antenna splitters, for two, three and four receivers, do not cause mismatches.

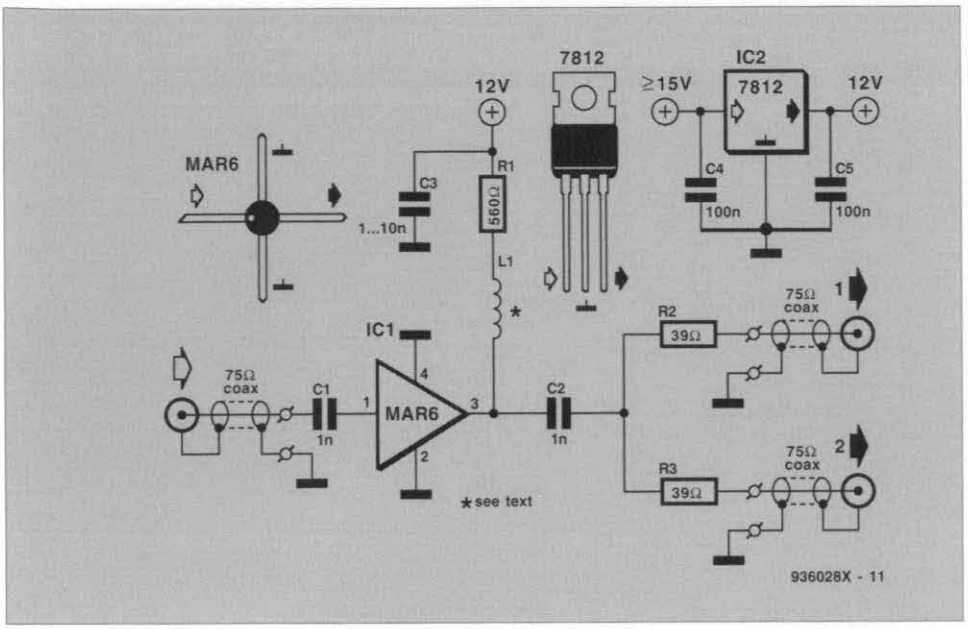


Fig. 4. Circuit diagram of the active antenna splitter.

Most modern FM receivers have a 300 Ω and a 75 Ω antenna input.

Apart from an impedance transformer, the impedance at both ends of the 75 Ω antenna cable can be put right by the use of a simple resistance network at the cable ends. The resistors used can be connected in series or in parallel with the impedance they are to correct. The network is designed such that whatever is connected to it, a TV receiver, an antenna or a coaxial cable, is always correctly matched.

The attenuation caused by the network, or by the division of the signal over several receivers, is remedied by the use of a wideband amplifier. This is done in the splitter discussed a little later.

Figure 2 shows a network as often used in cheap (passive) splitters. This is not a good setup, however, for, although the antenna 'sees' an impedance of 75 Ω, the receivers do not. The antenna is connected to two parallel-connected branches, each having an impedance of 150 Ω (75 Ω of the resistor and 75 Ω of the appliance). Each of the receivers, however, 'sees' a 75 Ω resistor in series with a parallel network. This network consists of the impedance of the antenna (75 Ω) and one of a 75 Ω resistor in series with the 75 Ω input of the other receiver. The impedance of the network is thus $(75 \times 150)/(75 + 150) = 50\ \Omega$. It is in series with one of the 75 Ω resistors, so that each receiver 'sees' an impedance of 75 + 50 = 125 Ω. Since the input of each of the receivers is 75 Ω, there is a serious mismatch.

Much better remedies are shown in **Fig. 3**. for, from left to right, two, three and four receivers from a single antenna or cable outlet socket. The impedances of the antenna or cable outlet and the receivers is represented by the 75 Ω resistor connected to the plug or socket by a dashed line. As stated earlier, these resistors ensure correct matching and prevent ghosting, but they do cause some attenuation.

Active antenna splitter

The attenuation caused by the splitters in Fig. 3 is nullified in **Fig. 4**, by the addition of a single operational amplifier, IC_1. Unfortunately, the output impedance of IC_1 is 50 Ω, which means that the resistance values of the splitters are different from those in Fig. 3.

The antenna signal at the input socket is applied to pin 1 of IC_1 via C_1. The amplified signal at pin 3 of IC_1 is applied to splitter resistors R_2 and R_3 and then, via 75 Ω cable, to the inputs of two (TV or radio) receivers.

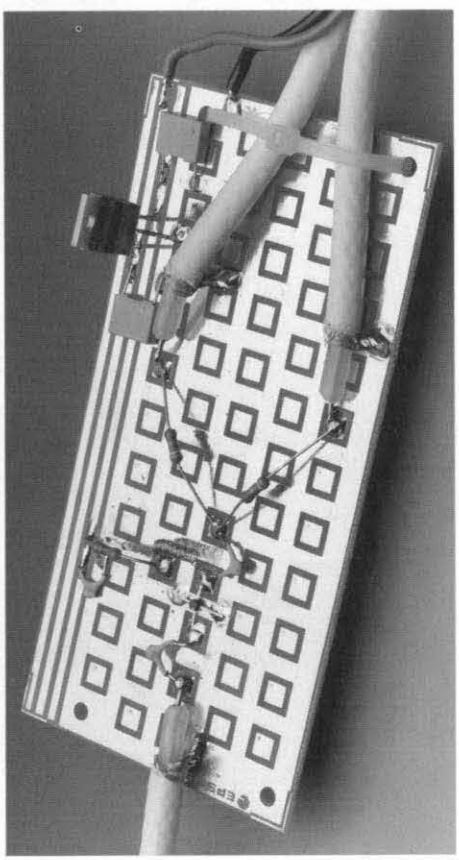
Power for IC_1 is supplied by a 12 V source via network R_1-L_1 to pin 3 of the op amp. Pins 2 and 4 are the negative or ground connections of the device. The 12 V is derived from a 15 V mains adaptor via regulator IC_2, which keeps the supply very stable, and thus free of interference.

The supply rail is decoupled by C_3, which at the same time prevents the connection between R_1 and the output of IC_2 picking up or radiating the antenna signal.

Capacitors C_4 and C_5 smother any tendency of IC_2 to oscillate and at the same time short-circuit any spurious signals.

Construction

The circuit may be built on a piece of prototyping board or on a ready-made board Type 85000 (see p. 70) as shown in **Fig. 5**. The layout shown is not obligatory, but it is important to keep the input and output of the circuit well separated. It is, of course,



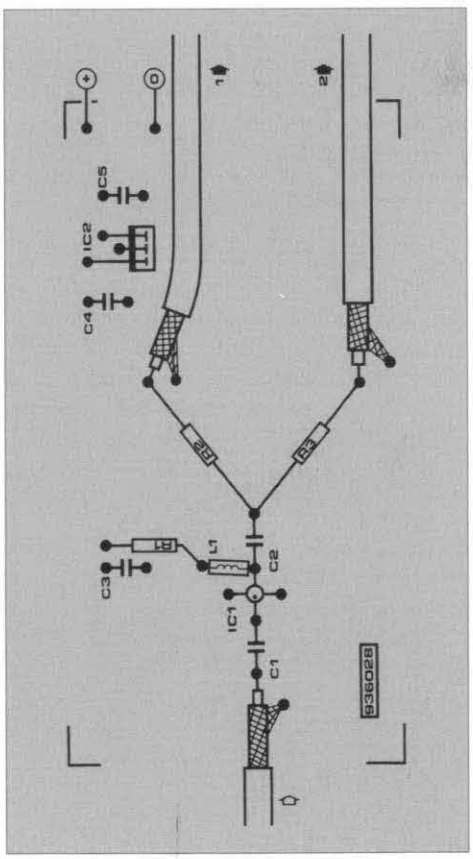


Fig. 5. Suggested construction on a Type 85000 board.

also possible to make the cable connections via plugs and chassis sockets instead of solder joints.

Choke L_1 consists of 15 turns (internal dia. about 4 or 5 mm) enamelled copper wire (0.2 mm thick).

Use

The splitter works best when it is connected directly to the antenna or cable outlet. If the signal is too strong (evidenced by multiple images on one or more channels), reduce the gain of IC_1 by giving C_1 a smaller value or by inserting a low-value resistor in series with the input.

The resistor network at the output of IC_1 works satisfactorily only if both outputs are terminated into 75 Ω. If one output is not connected to a receiver, it should be terminated into a 75 Ω resistor as shown in **Fig. 6**. This resistor can be soldered into a coaxial-cable plug as shown in **Fig. 7**. This arrangement makes it possible for a receiver to be connected now and then; when it is not, the terminating plug is inserted into the socket.

Parts list

Resistors:
R_1 = 560 Ω
R_2, R_3 = 39 Ω

Capacitors:
C_1, C_2 = 1 nF
C_3 = 1–10 nF
C_4, C_5 = 100 nF

Inductors:
L_1 = see text

Integrated circuits:
IC_1 = MAR6
IC_2 = 7812

Miscellaneous:
75 Ω coaxial plugs, sockets and cable as required.
Prototyping board Order No. 85000
[936028]

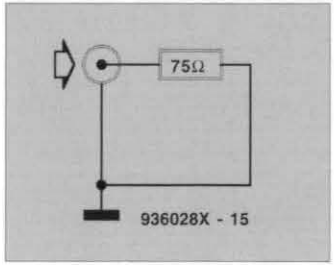


Fig. 6. Diagram of a 75 Ω terminating plug.

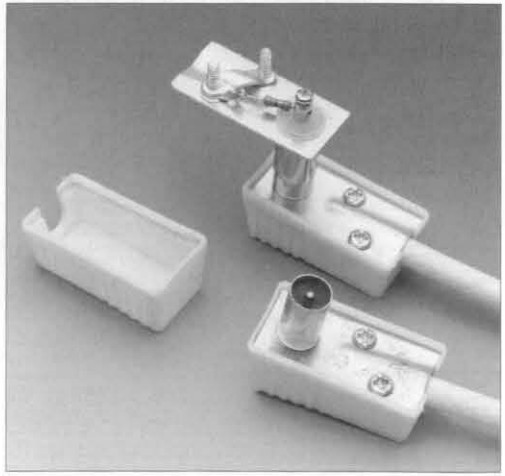


Fig. 7. How a 75 Ω terminating plug can be constructed.

CE LABELLING

Based on an article by R. Rastetter

Electromagnetic compatibility, EMC, is the property of an electrical/electronic apparatus to function satisfactorily without radiating unacceptable interference into its electromagnetic environment. At the same time, it should be immune to interference from other electronic/electrical apparatus.

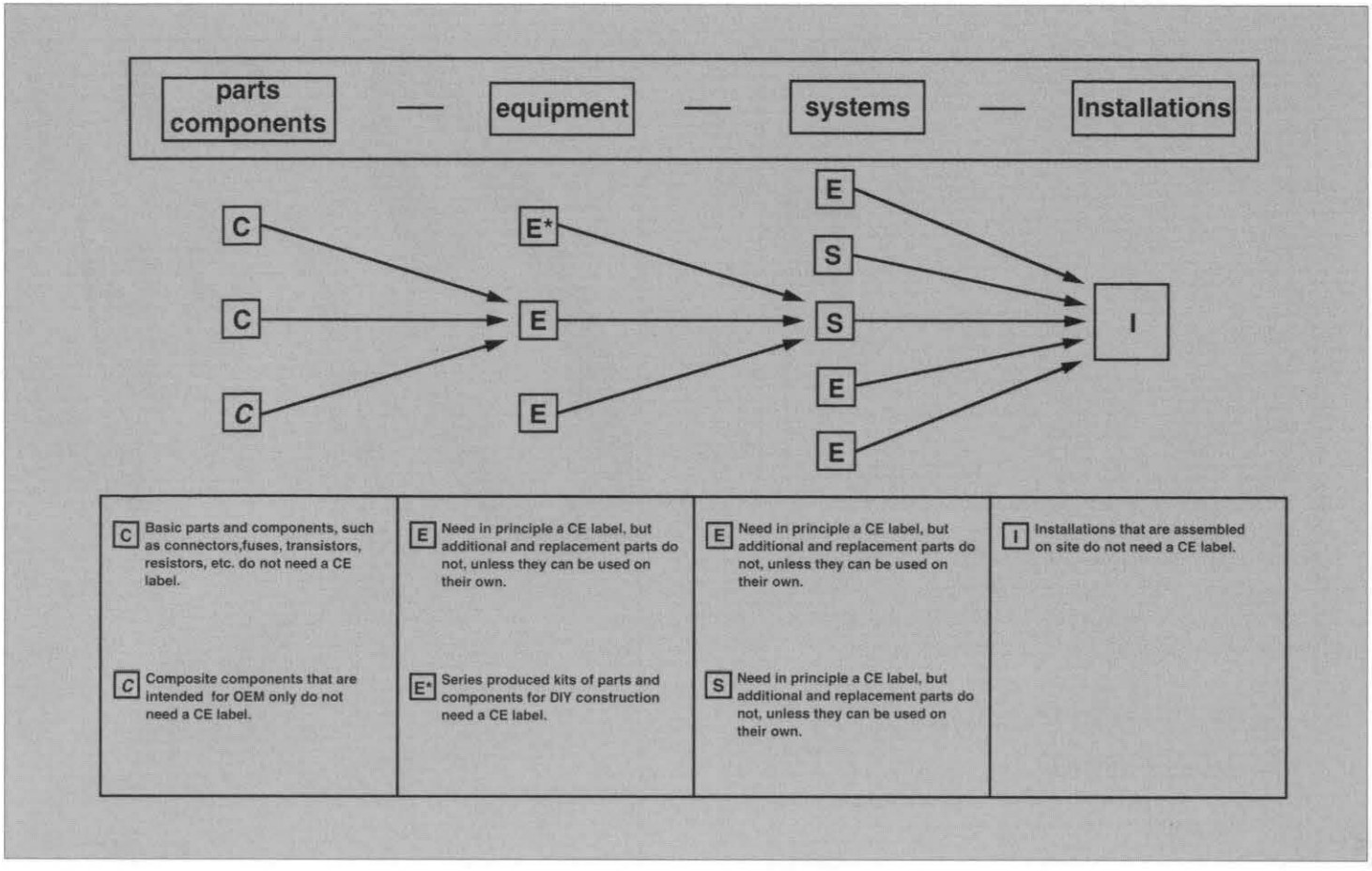
The member states of of the European Union (EU) have differing requirements as regards products that are sold and imported. This means that a product made according to the laws and regulations of one country can not be sold in other member states. However, one of the goals of the 1957 Treaty of Rome is the free movement of goods and services with the Union.

To this end, the European Commission issued a directive on EMC some years ago, which combines the requirements agreed by the member states as regards Public Health, Consumer Protection, Safety and the Environment. To indicate that a product complies with the stringent requirements of the directive, it must carry the CE (Conformité Européenne) label. This certification is, however, *not* intended as proof of quality to the consumer, merely as an indication that it complies with the requirements of the directive.

The EMC Directive (89/336/EEC) requires all electronic/electrical equipment that can cause interference or be affected by interference to comply with its guide lines. From 1 January 1996, all manufacturers must be able to show that their equipment complies with these guide lines. At the same time, all relevant electrical/electronic equipment may, from 1 January 1996, be marketed in the EU only if they are CE-certified (and also comply with any other relevant directives, of course).

When it is considered how much electrical/electronic equipment is in use in the EU (electrical tools, computers, lighting), it will be realised how much the directive will mean for interference-free use of such equipment in years to come.

As summarized in the drawing below, basic components, such as transistors, resistors, integrated circuits, connectors, and so on, are not subject to the directive and do not need CE labelling. However, composite components, such as thermostats, magnetic switches, or relays, do need a CE label *if they are marketed through the retail trade*. If these components are sold to OEM, they do not need a CE label. In general, according to the EMC directive, 'a component that has no intrinsic value to an end user' needs



no CE certification.

An electronic/electrical equipment comes into being by the assembly of modules, which in turn consist of basic and composite parts and components. Modules include, for instance, populated printed-circuit boards, monitors, computer cards, electrical household goods, and others. Whether such a module is encased or not does not matter in this context.

According to the guide lines, in the repair of a CE labelled equipment, CE labelled replacement parts or modules must be used. This ensures that the original EMC standards are maintained.

It is not entirely clear whether the directive has an effect on the amateur electronic constructor who builds an equipment on the basis of a magazine article. It is, however, certain that the user of equipment, even home-built, which interferes in whatever way with equipment of third parties is liable bylaw. Note also that kits of parts and components for DIY construction do need CE certification.

This magazine can not test the prototypes of all construction projects to the full EMC standard, and, therefore, we can not guarantee that any equipment built on the basis of an article does meet the standard.

From our January 1996 isse, we will regularly publish information to guide constructors how to build an equipment in accordance with the EMC standard.

At the same time, in the design we will take account of the EMC standard as much as is practicable. If necessary, an article will be accompanied by guidance on how ensure that a home-built equipment does at least comply with the EMC standard as far as interfering radiation is concerned.

There is as yet no clarification on the position of radio amateurs and their (often home-built) equipment in respect of EMC. No doubt, this will be forthcoming in the near future.

[950090]

THE DIGITAL SOLUTION

Part 10 (final) – Back to the real world

By Owen Bishop

In this series we look closely at digital electronics, what it is, what it does, how it works, and its promise for the future.

All the articles of this series have demonstrated that there is generally an advantage to be gained by storing, transmitting and processing data in digital form. Maybe there are disadvantages, too, and analogue data may sometimes be more easily handled, but the overall preference is for digital representation. This preference endures for as long as data is being handled by electronic devices. But ultimately, the data has to be presented to a human for assessment, as information, or even simply for entertainment, and humans operate in an essentially analogue manner. Maybe we can cope if the decimal number 12 is presented to us in binary form as 1110, but few would recognize that the number 111 1100 1011 represents the year 1995. Fewer still would recognize a certain sequence of bytes as one of the themes from Beethoven's Ninth Symphony. Before it can have much meaning to humans, digital data nearly always needs to be converted back to analogue form. In this final part of the series, we consider some of the many techniques of *digital-to-analogue conversion*.

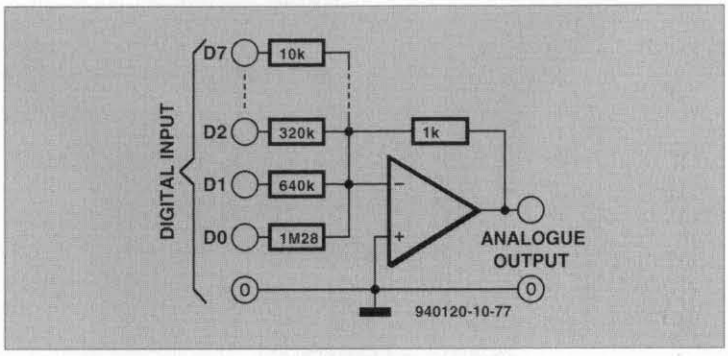


Fig. 78

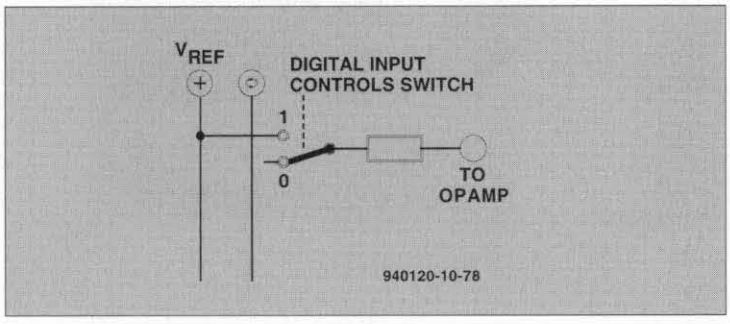


Fig. 79

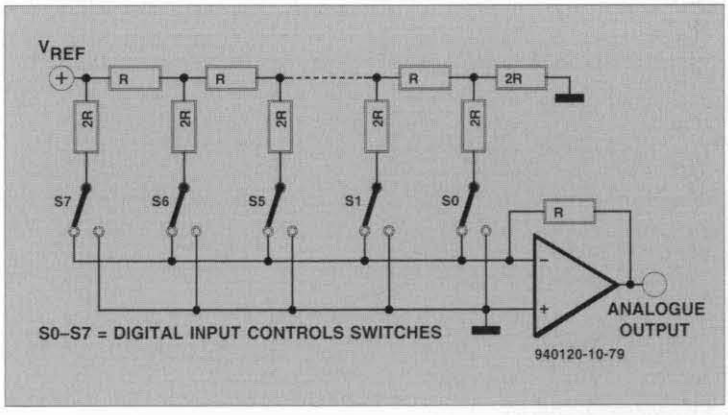


Fig. 80

Weighted resistors

One of the simplest DACs makes use of the summing abilities of an operational amplifier. In **Fig. 78**, the inverting input of the amplifier receives current from a number of sources through separate resistors. Assuming that this is an 8-bit converter, the digital value to be converted is applied to the eight inputs, D_0–D_7. The input is a logic level, either 0 V (equivalent to logic 0), or 5 V (equivalent to logic 1). The resistors are weighted, so that each one has a value half that of the one preceding it in the sequence. Correspondingly, the current flowing through each resistor when 5 V is applied to it is double that flowing through the preceding resistor. Thus, the currents are in proportion to the values of successive digits in a binary number. For example, the current flowing to the amplifier from D_0 is 5/1280000 = 3.90625 µA. If 5 V is applied to D_1, the current is 5/640000 = 7.8125 µA, exactly double (2^1 times). At the other extreme, the current flowing from D_7 is 5/10000 = 50 µA, which is 128 (2^7 times) the current through D_0.

In **Fig. 78**, the digital input is shown to be fed directly to the resistors. This is a workable circuit, but relies on the individual bits of the digital inputs all having precisely the same voltage levels for '0' and '1'. It is more usual for the digital input to be fed to an array of analogue switches, each connected as in **Fig. 79**. If the input bit is '0', the switch connects the resistor to 0 V. If the bit is '1', the switch connects the resistor to an accurate reference voltage, V_{ref}, which is used for all resistors.

This converter relies on the 'virtual earth' property of the inverting input of the op amp. Current flows to the inverting input as if this is at ground (0 V) potential. If two or more inputs are at 5 V, the current flowing through one resistor is independent of the current flowing through the others. As a result, the current reaching the inverting input is the sum of the individual currents. When two or more inputs receive a logic 1 input, the currents are summed, just as the value of a binary number equal to the sum of the values of the '1' digits it contains. Suppose both D_0 and D_1 are at 5 V. The sum of the currents is 11.71875 µA, which corresponds to the the binary value 11 (decimal 3) and is three times the current flowing when D_0 alone is supplied. If all eight inputs are high, the total current is 1111 1111 (255 decimal) times the D_0 current, or approximately 996 µA. The inputs of an op amp have extremely high impedance, so all the current flows on through the 1 kΩ resistor, creating a p.d. of 1 V across it. The output voltage swings *negative* of the inverting input by 0.996 V. As the binary input ranges from 0 to 1111 1111, the analogue output ranges from 0 V to –0.996 V.

The output is now on an

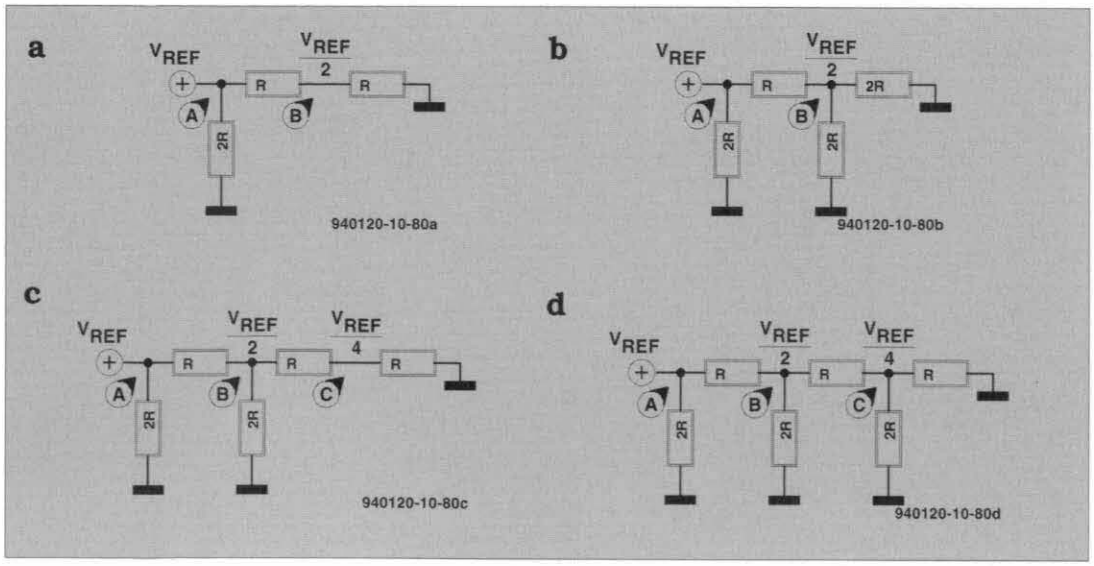


Fig. 81

analogue scale and can, if desired, be fed to a voltmeter calibrated from 0 to 255 to give a direct reading. More often it might be used to provide an analogue input to a device such as a motor, controlling its speed, or to a loudspeaker, determining the position of the cone. The output is said to be analogue, but this is not quite true. An analogue output could take *any* value in the range 0–0.996 V, but the output of this circuit is one of 256 steps, spaced 3.90625 mV apart. The *resolution* of the converter is 3.90625 mV. However, in many applications, the output is fed to a low-pass filter and, if the output is continually changing, the filter smoothes out the steps, producing an effectively analogue output.

This type of DAC is simple in principle and produces rapid conversions. Another version has the weighted resistors in the feedback position. These can be switched in or out of circuit by analogue switches controlled by the digital input. The input to the op amp is at a fixed level, but the gain, and hence the output voltage, can be set to 256 different levels. In either version, the chief disadvantage is that the values of all the larger resistors must be very precise. For example, if the 10 kΩ is precise to within 10% (±1 kΩ), *all* resistors must be precise to within ±1 kΩ. This means that the 1280 kΩ resistor must be precise to within 0.078%. A converter of more than 8-bit precision is not really a practical proposition.

A digital-to-analogue converter of this type in which the output is the product of the digital input and a reference voltage source is often known as a *multiplying DAC*.

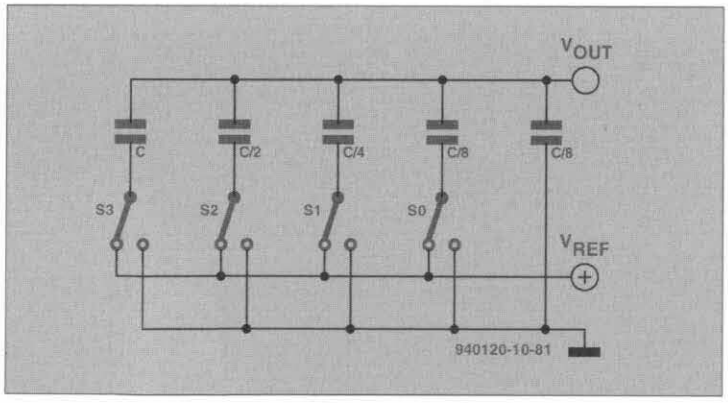


Fig. 82

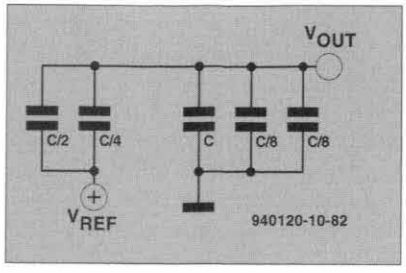


Fig. 83

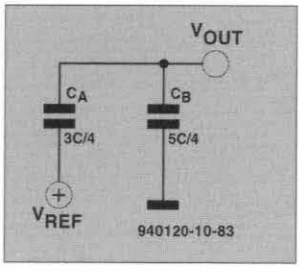


Fig. 84

Switched current-source DAC

This type of converter also uses the current-summing property of an op amp, but the circuit for producing the currents is entirely different from the weighted resistors used in the previous converter. In **Fig. 80**, the current sources are provided by what is known as an R-2R ladder, because it consists of resistors of only two values, R and 2R. Current flows from the voltage reference along the chain of R resistors (except the one at the end, which is 2R) and across the 2R resistors, which form the rungs of the ladder. The current flows through switches controlled by the digital input and then goes either to ground (if the switch is in position 0) or to the inverting input of the op amp (if the switch is in position 1). In either case, the current flows to a line at zero potential. If a switch is at position 0 (corresponding to a '0' in the data byte), the current is lost to ground. If a switch is at 1, the current flows to the op amp and is summed with other currents from switches at position 1. This is another type of multiplying DAC.

In **Fig. 81** we build up an R-2R ladder in stages, to see what potentials are present at its rungs. In the network of Fig. 81a, the potential at point A is V_{REF}. The two R resistors divide the potential exactly in half, so the potential at point B is $V_{REF}/2$. In Fig. 80b, the single R resistor on the right of Fig. 80a has been replaced by two 2R resistors in parallel, both of which are connected to 0 V. Two 2R resistors in parallel are exactly equivalent to a single R resistor, so the potential at point B remains at $V_{REF}/2$. In Fig. 80c, the 2R resistor at the right of fig. 80b is replaced by its equivalent, two R resistors in series. They divide the potential at B exactly in half, so the potential at point C is $V_{REF}/4$. In Fig. 80d, the R resistor at the right of Fig. 80c is replaced by two 2R resistors in parallel, and the potential at C remains at $V_{REF}/4$. This process can be repeated indefinitely, producing more and more 2R rungs, the chain of R resistors always ending in a 2R resistor tied to 0 V. At each rung, the potential is exactly half that of the rung above it. Since the 2R resistors are connected to 0 V whatever the position of the switches, the switch positions have no effect on the potential. When a switch is set to position 1, a corresponding current flows to the op amp. This current is $V_{REF}/2R$, $V_{REF}/4R$, $V_{REF}/8R$, and so on along the ladder to $V_{REF}/256R$ at the last rung of an 8-bit chain. It is feasible to continue the ladder to give 12-bit or 16-bit resolution. Whatever the length of the ladder, the op amp sums the current flowing through the 1 switches, producing an output ranging from 0 V to $-V_{REF}/R$ as the input (in an 8-bit converter) ranges from 0 to 1111 1111.

This is one of the most popular types of converter. It is

easy to produce in integrated circuit form, mainly because it requires on-chip resistors of only two values. More significant is the fact that it is not the *absolute* values of the resistors that matter, only their *relative* values. This is easy to achieve, allowing high precision at low cost.

Switched capacitor DAC

One of the newer forms of DAC relies on switched weighted capacitors. Instead of using a chain of resistors connected in series as a potential divider, these DACs use a ladder of capacitors connected in parallel. The principle of the technique is illustrated in **Fig. 82**, which for simplicity is ahown as a 4-bit DAC. The analogue switches are controlled by two clocks, which in turn produce high pulses. When clock 1 goes high, all switches are connected to the 0 V line, so all capacitors are discharged. Next, clock 2 goes high and this operates in conjunction with the digital input so that only those switches corresponding to a '1' input bit are connected to the V_{REF} line. The switches corresponding to a digital '0' remain connected to the 0 V line. However, this does not mean that they remain uncharged. To see what happens, suppose that the digital input is 0110. Switches D_0 and D_3 remain switched to the 0 V line, while D_1 and D_2 become switched to the V_{REF} line. It makes the situation easier to understand if the circuit is redrawn as in **Fig. 83**. The two capacitors on the left are those switched to V_{REF}. The three capacitors at the right are those switched to the 0 V line, and include the non-switchable C/8 capacitor at the end of the ladder.

The two groups of capacitors in Fig. 83 are capacitors in parallel, so they can be thought of as being replaced by a single capacitor for each group. Since they are in parallel, their equivalent values are found by adding their capacitances. **Figure 84** shows the equivalent circuit as a capacitive potential divider. It can be shown that when two capacitors C_A and C_B form a potential divider (Fig. 84), the output V_{OUT} is

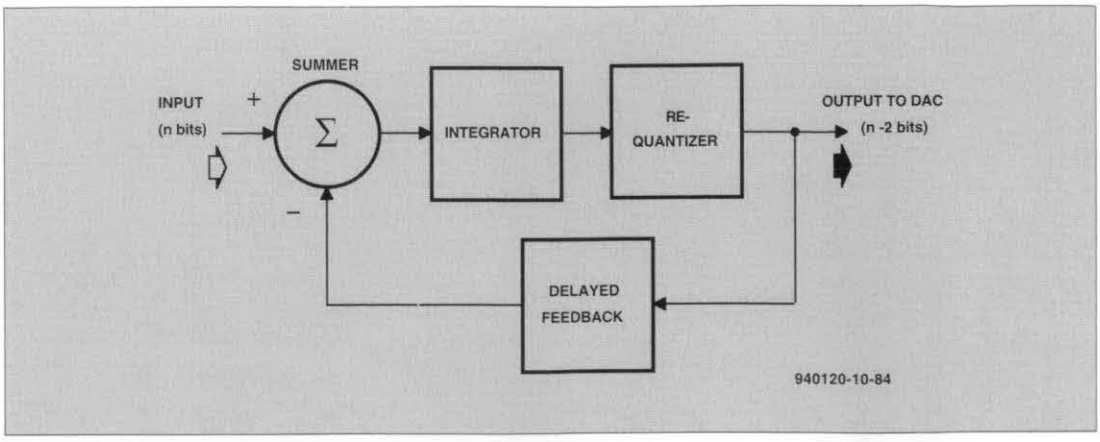


Fig. 85

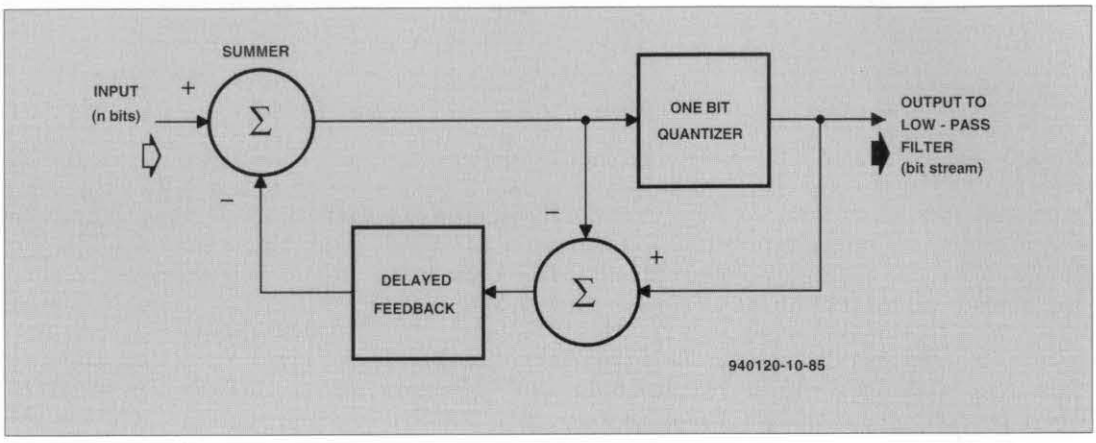


Fig. 86

$$V_{OUT} = V_{REF} \times C_B/(C_A+C_B).$$

Given that
$C_A = 3C/4$ and $C_B = 5C/4$,

$$V_{OUT} = V_{REF} \times [3C/4]/([3C/4]+[5C/4])$$

$$= V_{REF} \times 3/(3+5)$$

$$= V_{REF} \times 3/8.$$

The input is 0110 (decimal 6) and the output is 3/8 of V_{REF}. The relationship between input and output becomes clearer if we express the output in sixteenths, since there are sixteen possible switching combinations with four capacitors. Then, the output is written as 6/16. If the binary input is n, the output is $n/16$ of V_{REF}. This can be confirmed by working out the effects of other binary inputs. Note that since this DAC has no op amp, the maximum output is V_{REF}.

Using fewer bits

We have mentioned the problem of obtaining the required degree of precision in the most significant bits. An entirely different approach is to rely on fewer bits but to take samples more frequently. Taking samples more frequently than is required to represent the analogue data accurately is known as *oversampling*. For example, digital audio requires sampling at a little over 40 kHz (usually 44.1 kHz) in order to represent all frequencies in the audio range up to 20 kHz, the highest frequency that can be detected by the human ear. Sampling at a higher rate, such as 88.2 kHz, is oversampling. It does not improve the reproduction of audible sound, though it has the advantage that it is easier to filter out the quantization noise which results from the fact that, as mentioned earlier, the output from a DAC is stepped, not continuous. The frequency of this noise extends around the sampling frequency, so is much easier to remove when the sampling frequency is higher, well separated from the audio frequencies. There are practical problems in taking advantage of oversampling when producing CDs,

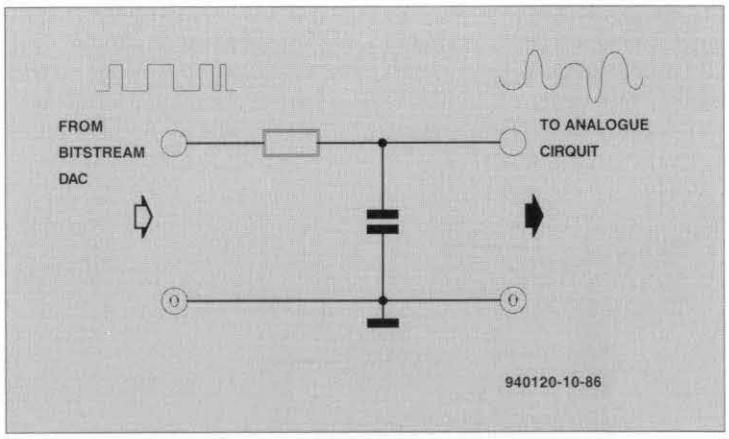


Fig. 87

but it is possible to effectively double the sampling rate on replay by interpolating a sample between each of the recorded samples. The value of each interpolated sample is calculated as the mean of the two recorded samples on either side of it. Interpolation was widely used in earlier digital audio systems, but is now largely replaced by systems that rely on fewer bits.

Figure 85 is a block diagram of a circuit that reduces the number of bits in the samples by 2. The digital input consists of samples, each of n bits, taken at the sampling rate f (say, 44.1 kHz). The sample is fed to an integrator, which digitally adds the sample to the value already there. The sum is then requantized, that is, the final two bits are removed from it. In effect, it is divided by 4 and the remainder discarded. This becomes the output of the circuit. The output is also fed back and subtracted from the incoming signal to compensate for the loss of the remainder. The key to the action of the circuit is that it operates at four times the sampling frequency, so that each input sample is held for four cycles. Note that this process is entirely digital. The output from this circuit may then be fed to a DAC with two fewer bits, and hence greater precision than would otherwise be required.

A similar principle may be used to build a 1-bit DAC as shown in **Fig. 86**. In this, the quantizer* has only two levels of output, '0' and '1', and is known as a 1-bit quantizer. If it receives a sample with a value up to but not including 0.5 of the maximum value, its output is '0'. If the sample has a value of 0.5 or more of the maximum value, its output is '1'. As before, the operation is repeated several times for each input sample. However, in this case, we do not feed back the output signal, but the difference between the output signal and the sample, so as to allow for the loss of information stored in the discarded bits. To see how this works and the nature of its output, follow the stages in converting a 4-bit digital value, reducing it to 1-bit output. Given that the 4-bit input has 1000 (decimal 8) as its maximum value, the quantizer gives an output '0' for values 0000–0011 and an output '1' for values 0100–1000. Because it drops three bits, the DAC cycles at 8 ($=2^3$) times the initial sampling frequency. What happens is shown in Table 8.

At the first cycle, there is no existing feedback; the total is between 0100 and 1000, so the output is '1'. This '1' is really a '1' plus three following bits which have been discarded, so it is equivalent to 1000. This is subtracted from the input+feedback value to give the feedback (–0011) for the second cycle. In the second cycle, the feedback is added to the input (which is held at the same value as before). The feedback value is negative so we show this as a subtraction, giving 0010. This is between 0000 and 0011, so the output is '0'. Subtracting this (=0000) from 0010 leaves +0010, which becomes the feedback for the third cycle. This process is repeated until, at the end of eight cycles, the feedback is zero. The input has been completely converted into a series of 0s and 1s without any remainder.

In the above example, the input value is decimal 5, and the output has five '1' bits in it out of eight bits. Thus, the output is 5/8 of the possible maximum of 8. The reader might try this routine for other inputs to confirm that, for any input between 0000 and 1000, the number of '1's produced in eight cycles always equals the value of the input. This procedure can be extended to any number of input bits and always the proportion of '1' bits in the output equals the proportion of the input value to the maximum value.

The output of a DAC such as this is a stream of '1' bits (high logic level) interspersed with '0' bits (low logic level) and is known as a *bit stream*. The bit stream is produced wholly by digital circuits, which means that there are no problems with precision, ageing of components, noise, or any of the other snags that beset analogue circuits. It might be objected that the signal is still digital after it leaves the converter, but all that is now needed is a low-pass filter of simple design to convert the stream of bits (remember, these are coming at a very high rate) into an analogue signal. No further *conversion*, as such, is needed. Assuming, for the sake of simplifying the explanation that we use the simplest possible kind of low-pass filter—see **Fig. 87**, each '1' pulse places a unit charge on the capacitor. A continuous series of '1's rapidly charges the capacitor to the maximum voltage and the output signal reaches this level. With a continuous series of '0's, no charge is supplied to the capacitor and the output rapidly falls to zero. In the intermediate situations in which between one and seven pulses are produced for each sample, the capacitor is charged to a greater or lesser degree and an intermediate output voltage is generated accordingly.

Bit stream technology is one of the newer developments of digital electronics and has special applications in audio electronics such as in CD players. It has its limitations, the primary one being that to convert a 16-bit sample requires a bit stream DAC cycling at about 3 GHz. This is not easy to achieve in the present state of circuit fabrication, so hybrid techniques are used, such as reducing the number of bits by using circuits such as that in Fig. 85 and completing the process with conventional DACs, or combining bit stream techniques with interpolation.

* to quantize = to digitize = to convert an analogue signal to a digital signal.

Input + feedback	Output	Feedback
0101+0000=0101	1	0101–1000=–0011
0101–0011=0010	0	0010–0000=+0010
0101+0010=0111	1	0111–1000=–0001
0101–0001=0100	1	0100–1000=–0100
0101–0100=0001	0	0001–0000=+0001
0101+0001=0110	1	0110–1000=–0010
0101–0010=0011	0	0011–0000=+0011
0101+0011=1000	1	1000–1000= 0000

Table 8

Converted?

In this series we have described and explained many aspects of digital electronics and we have attempted to show that digital techniques have many benefits. Analogue techniques have a lot to offer and may be preferable in many situations but, on the whole, the author believes that ultimately the practitioners of electronics, professional and amateur, will be drawn increasingly toward *The Digital Solution*.

[940120-X]

Answers to Test yourself (9)

1. Active high.

Q_n	R	S	Q_{n+1}
L	L	L	$\bar{L}$
L	L	H	H
L	H	L	L
H	L	L	H
H	L	H	H
H	H	L	L

2

Latch No 1	2	3	4	XOR (of 3&4) fed back	Output (4)
0	1	1	0	1	0
1	0	1	1	0	1
0	1	0	1	1	1
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	0	1	0
1	1	1	1	repeats	

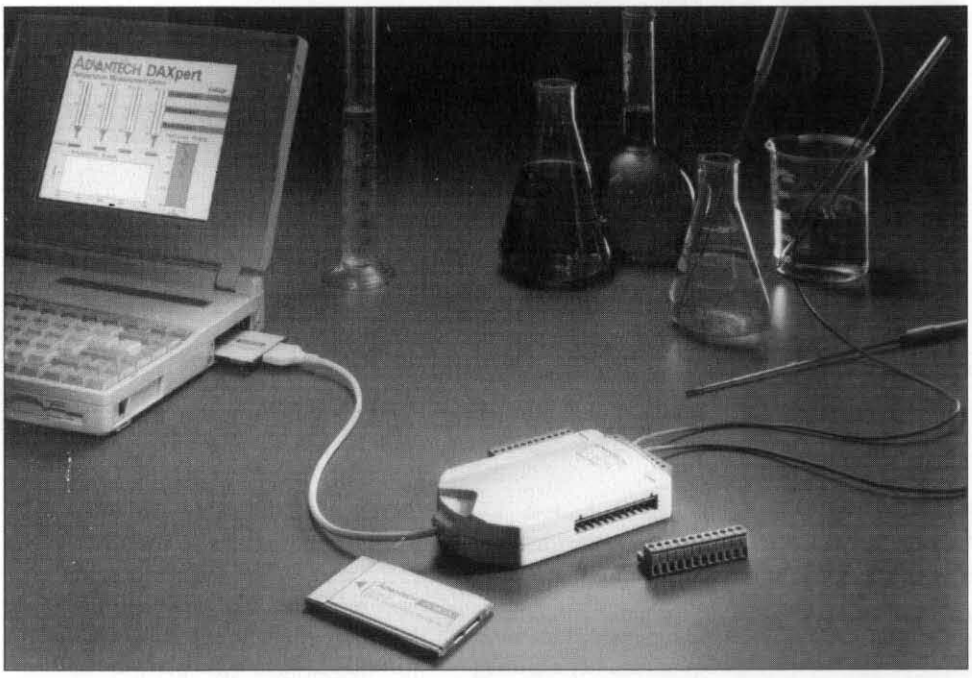
3a 4095

3b 16777215

FOCUS ON: MEASURING WITH THE PC

AN INSERTION CARD FOR EVERY TYPE OF MEASUREMENT

More than twenty years ago, IBM designed a computer system for small office applications. No one at that time could have foreseen the revolution the PC (personal computer) would bring about. This article describes how the today's computer can be put to use in the measurement of electrical quantities, an application which is totally different from that for which the PC was originally developed.



Photograph courtesy Advantech.

By our editorial staff

FOR many applications, the times are gone when measuring just about any electrical quantity called for moving coil instruments. In an increasing number of cases, it is found that a digital instrument is the most practical and accurate solution to analyzing a wide range of quantities. The use of a computer is the next logical step if the measurement results are to be processed numerically, if any kind of calculation is required, or if the results are to be included in a document. Before that can be done, however, the information has to be measured and read into the computer's memory. That can only be done with the aid of special interfaces. So, what types of insertion card and interfaces are available for this purpose, and what functions do they offer?

Function of the PC

Traditionally, there are two ways of incorporating a PC into a measurement set-up. The oldest is an arrangement by which the computer issues commands, via a special bus, to 'intelligent' measuring instruments. The most popular bus for this type of setup is the GPIB (General Purpose Interface Bus), defined in IEEE488.2. Designed by Hewlett Packard, this bus was originally introduced as the HP-IB (Hewlett Packard Interface Bus) as long ago as 1965. Applications of the HP-IB remain limited to driving (controlling) a complete instrument. So, a computer having such a bus is not capable of performing measurements itself.

A single GPIB interface allows up to 15 instruments to be controlled. The popularity of the GPIB is proven by the fact that there exist more than 3,000 test instruments having this type of interface, including oscilloscopes, power supplies, spectrum analysers, scanners and multimeters. That makes the GPIB the interface with the widest support in the field of electronic test and measurement equipment. Interface cards that add a GPIB interface to a PC are available in many different versions, even including compact PCMIA cards. GPIB converters are also available for connection to the RS232 or Centronics printer port. These converters allow even ordinary laptop PCs, which do not have room for insertion cards or PCMIA add-ons, to be upgraded with GPIB control, and used in an intelligent measurement setup.

GPIB extenders are available as options to enable the interface to cover distances between 20 m and 1 km without problems. Using this approach, the highest possible transmission rate is about 2 MByte per second.

These days, there are a number of programs which enable instruments to be controlled, via the GPIB bus, from Visual BASIC, Visual C++, or any other Windows-based software development environment.

What can I measure, and how?

Among the advantages of using a PC as a measurement instrument are its reliability and accuracy. Computer-controlled measurements allow the time of the measurement and the interval between two measurements to be set with great accuracy. Moreover, measurements can be made around the clock, 24 hours a day, seven days a week. This ability plays an important role with measurements that are necessary to investigate the trend of a process over a relatively long period of time, or with measurements which are performed in difficult to reach locations. Another important advantage of using a PC for measurements is that all measurement results are available in digital form. Consequently, these data are easily 'put on file', or archived. Another possibility is to send data to a central location for analysis, via a network or

modem line. Moreover, measurement data collected by a PC-based instrument may be converted (if necessary), and used in existing programs such as databases, spreadsheets and word processors. Specialized software is available to perform complex operations on the data, such as FFT analyses.

The interfaces currently available may be divided into two groups: cards for analogue signals, and cards for digital signals. In many cases, a combination of these is also available.

Measurements on digital signals (TTL or CMOS) are often supported by additional counters, dividers and timers which are implemented on the card. A deep analysis of the operation of a complex logic circuit, for instance, a microcontroller system, is only possible with purpose-designed hardware. Most 'ordinary' measurement cards have only a limited application area such as counting pulses (for example, the number of times a valve closes), detecting logic levels (establishing the position of a valve), or measuring frequencies (running speed of an engine).

Processing analogue signals is a different kettle of fish. Because most measurement cards accept input voltages, special transducers have to be used to convert a quantity to be measured (for instance, temperature) into a corresponding voltage. Other quantities that may have to be measured include gas or liquid flow, pressure, acidity, electrical current, strain and force.

After the quantities have been converted to electrical voltage levels, these are translated into digital codes by an analogue-to-digital converter. The width of the code is usually between 8 and 16 bits. Smart algorithms are capable of filtering these signals, and eliminate small irregularities. Next, the values of these 'cleaned' signals are read by the system. Depending on the software used, the system may actually be measuring amplitude, frequency, effective level, distortion or a whole frequency spectrum.

How the measured and pre-processed information is offered to the user depends on the applied software. Sometimes the presentation takes the form of a complete, simulated, oscilloscope or spectrum analyser. In other cases, the presentation is limited to the measured value only, which is displayed in a small window on the screen.

External and internal measurement systems

In principle, there are two ways of connecting measurement circuits to a PC. Either an internal expansion slot is used, or one of the standard interfaces on the PC, i.e. RS232 (serial) or Centronics (parallel). Insertion cards can reasonably be expected to offer the best performance as far as speed is concerned. On the other hand, measurement systems which are connected to one of the standard ports are easy to connect, and may be moved between PCs.

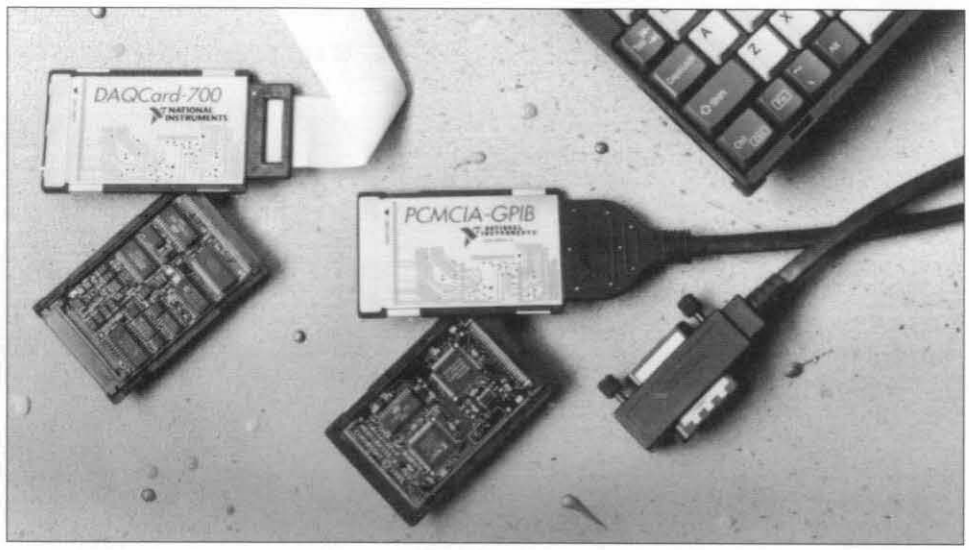


Fig. 1. This compact interface card enables a PCMIA slot to be converted so that it can be used to control measurement instruments via the GPIB interface.

Any reasonably fast PC (we're not talking of XTs here) having a free ISA slot may be turned into a digital oscilloscope. The type of interface to be purchased will depend mainly on the available budget and the desired performance. Rather than providing an overview of technical specs of all related products currently on the market, this article shows a number of options to indicate the various possibilities.

An example of a relatively inexpensive external instrument is an intelligent multimeter having a serial interface. Thanks to this interface, the multimeter is able to copy its measured data to a computer. The latter does not really partake in the actual measurement, but only receives the data in the form of a file. Keys on the instrument itself determine its settings, and the start of the measurement. This makes it clear that such a meter is not suitable when the computer is to record the complete measurement process.

What can I buy?

The product range offered by National Instruments clearly indicates that this company is committed to measuring and controlling with the aid of a computer. National Instruments supply interfaces and software for many computer systems and platforms, including PC, Apple Macintosh, SUN and NEC.

An example of an advanced acquisition system is the new AT-MIO-16E02

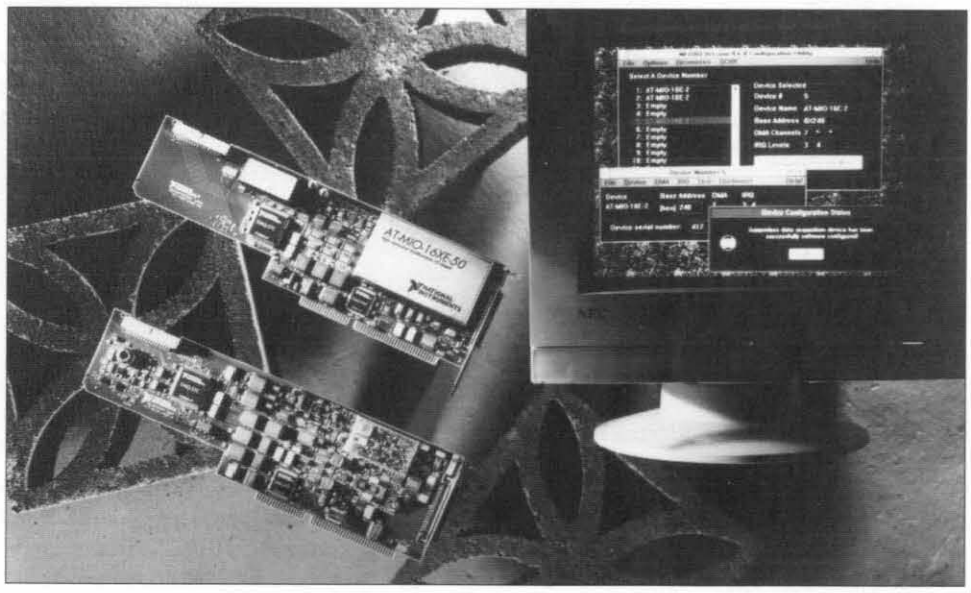


Fig. 2. This card from National Instruments is a multi-functional data acquisition system. It is capable of processing analogue as well as digital signals.

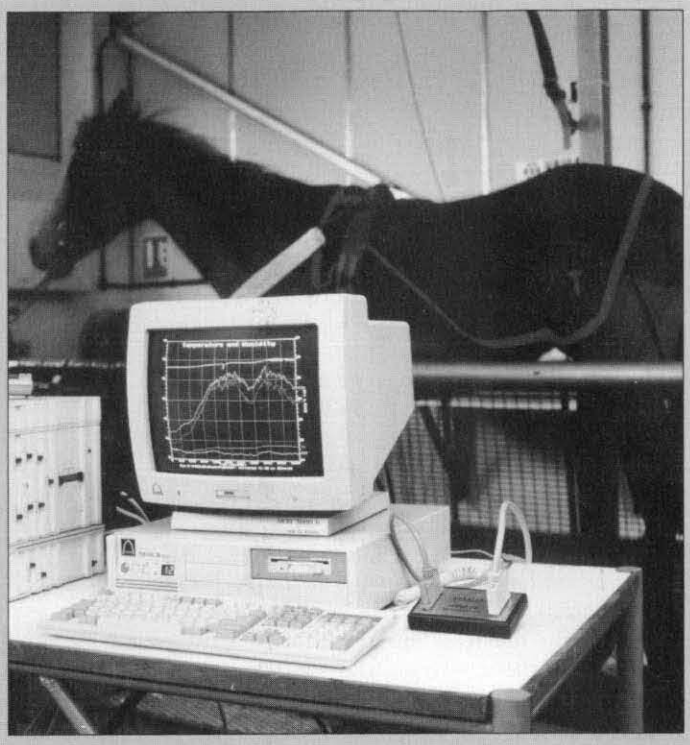
Pico ADC Virtual Measurements

Pico was one of the first companies to release hardware and software specifically designed for use as virtual instruments, and has remained a leader in the field of computer measurement with its comprehensive range of products.
Pico ADC units plug into either the computer's parallel or serial port, and require no expansion slots or power supplies. The use of these interfaces is becoming increasingly common for data acquisition equipment as it allows easy connection, and is currently the only method available that allows the same hardware to work on both desktop and laptop computers.
The range includes units that are optimised for datalogging use such as the high-resolution (16-bit + sign) ADC-16, and the 22-channel ADC-22. Other units have been designed for use as virtual instruments such as the dual channel ADC-100 which is ideal for use as an oscilloscope. There are also specialised units such as the TC-08 for thermocouple measurement, and the SLA range of logic analysers.
Two main software packages are supplied for use with the units: PicoScope Virtual Instrument software (for using your PC as an oscilloscope, spectrum analyser or meter) and PicoLog data logging software.
One of the most popular units is the ADC-100 which has been described as 'a complete electronics lab in your PC'. With the supplied PicoScope software, you can use your PC to emulate the following instruments:

- dual channel storage oscilloscope
- XY mode oscilloscope
- spectrum analyser (0 to 50 kHz)
- voltmeter (ac/dc or frequency)
- frequency meter (0 to 30 kHz)

The ADC-100 offers both a high sampling rate (100 kHz) and a high resolution of 12 bits. Nine input ranges, providing ±50 mV to ±20 V full-scale enable the unit to be directly connected to a wide range of signals.
As well as a general-purpose lab instrument, the ADC-100 has proved popular in many diverse fields. For example, they are used in garages for tracing faults in car ignition, fuel injection and ABS circuits. In addition they also prove invaluable to audio engineers as the spectrum analyser covers the whole audible range.

Olympics provide a testing time
During the 1992 Barcelona Olympics several horses suffered from heat exhaustion. Climatic conditions in the 1996 Atlanta games will be even more severe, and there has been talk of the equestrian events having to be



modified or even cancelled. The Animal Health Trust in Newmarket has set up a project in order to make recommendations on transport, acclimatization periods and modifications to the competition required to ensure that the horses are not placed under unnecessary stress. The trust has set up an air conditioned building so that horses may be exercised on a treadmill under varying conditions. Pico virtual instruments and data loggers are being used to measure temperatures, humidity, sweat rate, oxygen levels and heartbeat signals.
The latest tests will be using the new TC-08 eight-channel thermocouple converter. By measuring the temperature at eight points on the horse's body, a 'weighted index' of the horse's body temperature will be constructed. The TC-08 will allow these tests to be performed far quicker than with traditional instruments.
During the Olympics, Pico ADC units will be used to monitor shade temperature, relative humidity and black globe temperature (an estimate of solar radiation). A cut-off level will be set above which it is not considered safe for horses to continue. If the index reaches this level, the competition will be temporarily halted until it becomes cooler.

(**Fig. 2**). This is a multi-purpose, multi-I/O card that can be inserted into an ISA slot which should be available in almost any PC except portables etc. The card offers a choice between 16 single-ended or 8 differential inputs, 8 digital I/O channels and 2 counters/timers. The maximum sampling rate is 500 kHz, and the two A-D converters used on the card achieve a resolution of 12 bits. The inputs of the card have a span of either ±5 V or ±10 V. A linear measurement range of 0 to 10 V is available. Moreover, the gain in each channel may be set to 1, 2, 5, 10, 20, 50 or 100.

The card is compatible with different programs, both running under DOS and Windows. For this purpose National Instruments supplies the programs NI-DAQ, LabVIEW and LabWindows.

To mention another example of an essential technical characteristic of an insertion card, the oscilloscope card from TiePie Engineering (**Fig. 3**) has a sampling frequency of 50 Msamples/s coupled with a resolution of 8 bits. Functionally, this card turns the PC into a digital memory oscilloscope with two input channels. The associated software incorporates all functions which are necessary to enable the use as a spectrum analyzer, a digital true-RMS multimeter for signals to 5 MHz, a distortion meter and a transient recorder. A built-in signal generator supplies a square wave output signal with a frequency between 100 Hz and 1 MHz. The resolution of 8 bits is typical of many measurement cards. In practice, it is adequate for a large number of applications. On the screen, it is usually not necessary to display more than 256 levels. Portable digital instruments (for example, the

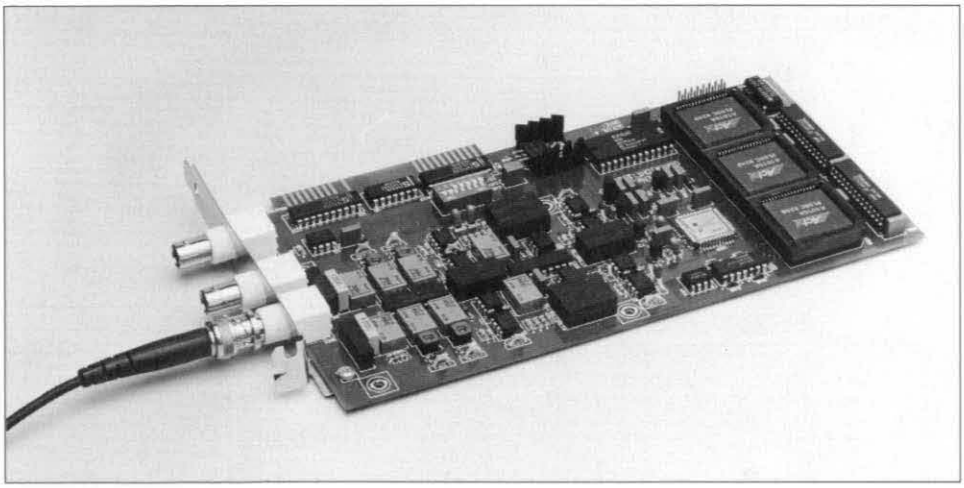


Fig. 3. The TP508 from TiePie Engineering turns any PC into a fully-fledged oscilloscope. Moreover, the software that comes with the unit offers a host of extra functions.

Tekscope from Tektronix) have a comparable resolution. However, if the measured data is to be used in complex calculations, a resolution of 12 bits is recommended.

An example of a simple and affordable solution is the MC-PC20 measurement card from Conrad Electronics (**Fig. 4**). This card demonstrates that functionality can be upheld despite a somewhat simpler design. This card offers 16 single-ended or 8 analogue inputs. The inputs feature overvoltage protection up to about 30 V, and are suitable for measuring voltages in three ranges: between ±10 V, ±5 V or ±2 V. Both the input and the output of the converter have a resolution of 12 bits. In addition to the analogue inputs, the card also offers 32 digital I/O channels at TTL level which can be configured as inputs or outputs in blocks of eight. The MC-PC20 comes with a large number of drivers.

Still more compact

Probably the simplest way to create an analogue input on a PC is to connect a compact adaptor to the parallel port. Such an adaptor enables portable computers, which have no extension slot at all (such as ISA, PCI or PCMIA), to be employed for simple measurement jobs. The ADC-12 from RS Components shown in **Fig. 5** is an example of such an adaptor. The unit has a resolution of 12 bits, an input range of 0 to 5 V, and a highest sampling rate of 18 kHz. A 10-bit version is also available offering a maximum sampling rate of 22 kHz. The input of both adaptors is a formed by a BNC socket. The ADC-11, finally, is a rather special adaptor which turns the printer port into 11 analogue inputs with a resolution of 10 bits each. An additional advantage of this adaptor is the presence of a digital output which can be used to control other circuits, or a heavier load, via a relay.

Two programs are available to give software support to these adaptors. PicoLog allows you to watch an input voltage over a long period of time, and store the measurement results. This function is normally referred to as logging. PicoScope turns the PC into a (simple) analogue instrument with the following functions: memory oscilloscope, spectrum analyser, voltmeter and frequency meter.

Another converter, the ADC-16, is connected to the RS232 (serial) port on the PC. This is basically an analogue interface with a resolution of 16 bits and up to eight channels. At a resolution of eight bits, a sampling rate of 200 per second is achieved. This is reduced to 2 samples per second when a resolution of 16 bits is selected.

As already mentioned, there are also interfaces available for extremely compact computer systems such as laptops and palmtops. Advantech supplies a PCMIA card (type II) with 12-bit resolution for these computers. The acquisition system, DACpad-71, offers eight differential inputs which are read at a sampling rate of 30 kHz. In addition to eight analogue inputs, four digital inputs and outputs are available. Depending on the exact type, the gain of the DACpad-71 may be set to a maximum of 8 (type 71A) or 1,000 (type 71B). A separate connec-

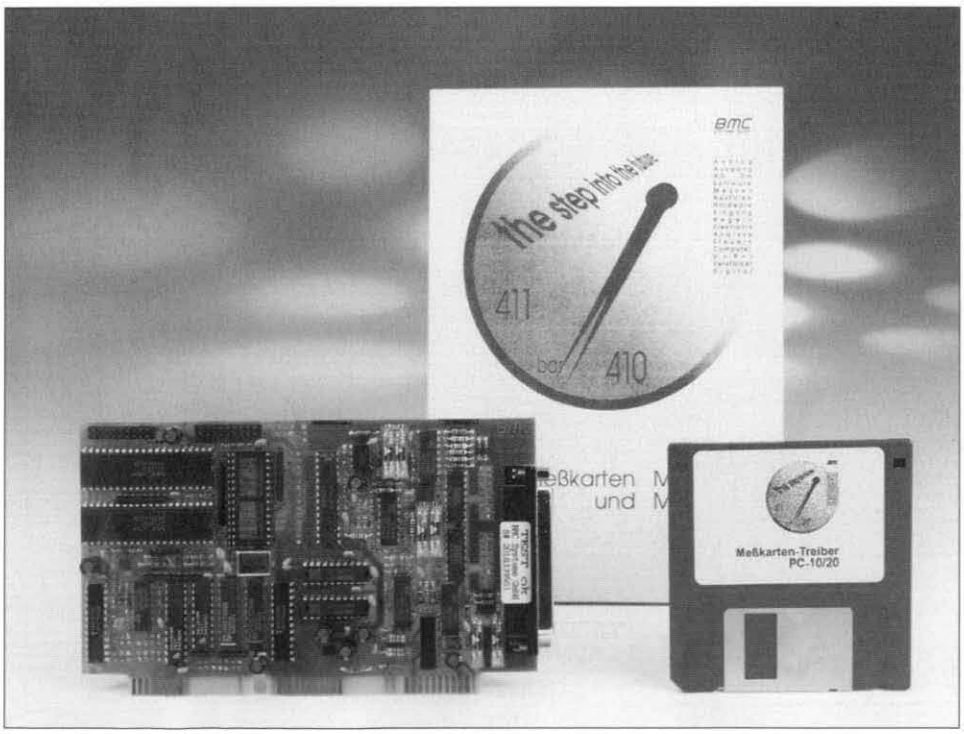


Fig. 4. This interface card from Conrad Electronics clearly shows that the simple, inexpensive, alternative does exist.

Fig. 5. This adapter from RS Components enables laptop and palmtop computers having only the standard interfaces (such as printer port) to be used for measurements. Great for field work and servicing! The software is supplied by Pico Technology.

tion box contains a cold junction compensation, which allows the interface to be used in conjunction with thermocouples.

Software

Much hardware comes with the associated software. The best known supplier of measurement and control software is, without doubt, National Instruments, whose programs are among the most frequently used in the relevant branch of the industry. The programs LabVIEW and LabWindows have a multi-purpose structure, and are capable of using a library with drivers for hardware produced by no fewer than 45 manufacturers. These VI libraries contain drivers for instruments which are connected to the GPIB, VXI or RS232 interface, but also for a number of insertion cards. Thanks to this support, you need not slave away at low-level programming to develop your own driver.

LabVIEW has a long history which goes back to 1983 when this development system was initiated, followed by the introduction on the market in 1986. In 1990, important patents were obtained on this software, resulting in the introduction of LabVIEW 2. Starting with this version, a graphics compiler is employed which results in a processing speed which is roughly equal to that of code developed in C. Later, LabVIEW was made suitable for use on computers on different hardware and software platforms: Microsoft Windows (NT), (Power) Macintosh, Sun Solaris and HP-UX.

The graphics environment offered by LabVIEW enables you to build an application which, in its final version, represents all the functionality defined by the user. Via LabVIEW, support is available for data acquisition, control and data analysis. Raw data collected with the aid of the software and the measurement instrument may be processed using DSPs, digital filters and statistical and numerical analysis.

An important advantage of the program is that it keeps using the same intuitive user interface over and over again, so that the user and the software always communicate in the same way.

Alongside the LabVIEW graphics environment there also exists LabWindows/CVI, an interactive environment which makes use of ANSI C. This integrated C environment contains, among others, a 32-bit C compiler with linker and debugger, and a couple of utilities for the generation of code. A staggering 450 drivers are available to support measurement systems from various manufacturers, which is a great help for any programmer involved in writing software. Included with LabWindows/CVI is a library containing routines for signal processing, statistics, curve fitting, FFT and complex analyses. A special graphics editor is available for the design of graphics blocks via which the software communicates with the user. A large number of icons is included to compose a nice looking screen, showing panel meters, switches, controls and indication lamps. Network support is available via DDE and TC/IP routines.

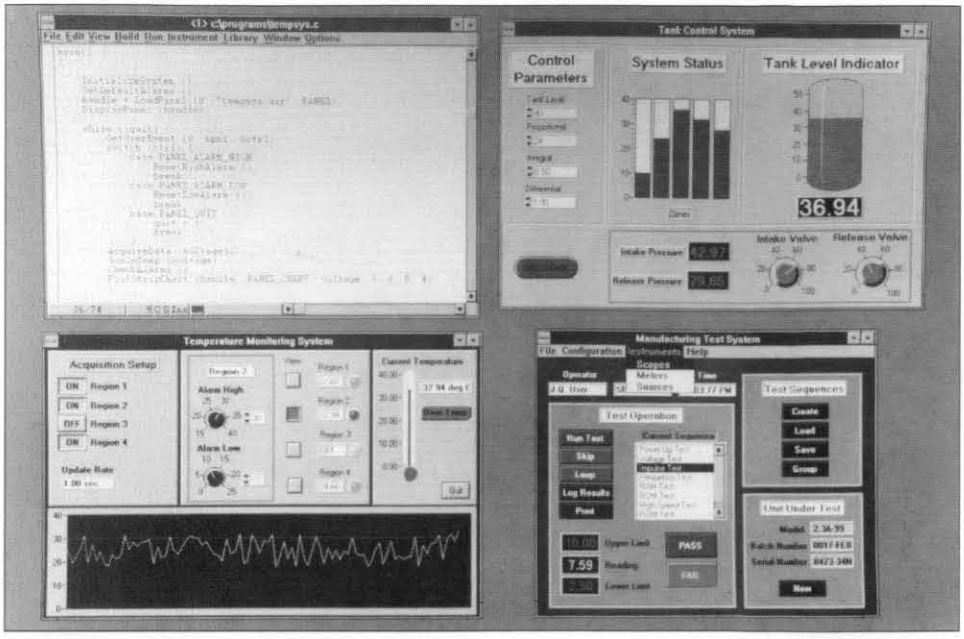


Fig. 6. LabVIEW offers the user a graphics environment for the production of made-to-measure applications.

Many possibilities

This article has only skimmed the hardware and software options which are available to perform and process measurements using a PC. A cautious estimate is that there are currently hundreds of external measurement systems for the computer, making it an impossible task to give a complete overview of all the technical features. In any case, the aim of this article is to give you some idea of the possibilities. And there are quite a few! Perhaps it would be better to make a (short) list of things that can not be measured with the aid of a computer?

(950081)

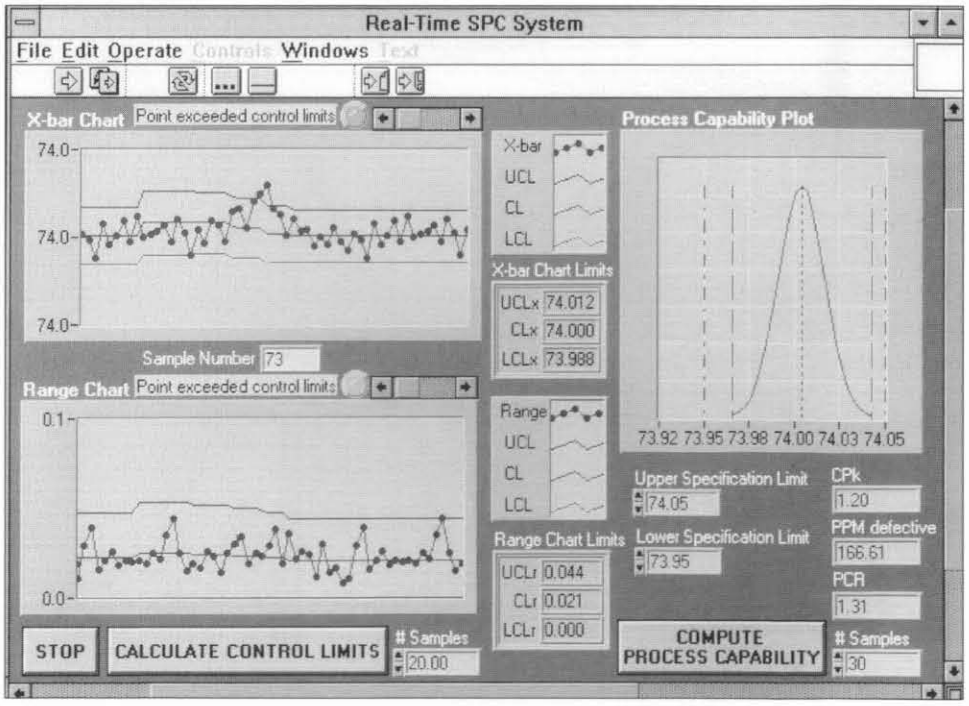


Fig. 7. LabWindows/CVI is an interactive software development system for the design of instrumentation software in an ANSI C environment.

APPLICATION NOTE

The content of this note is based on information received from manufacturers in the electrical and electronics industries or their representatives and does not imply practical experience by *Elektor Electronics* or its consultants.

Signal-line circuit protectors

A Maxim Application

The MAX366 and MAX367 are multiple, two-terminal circuit protectors. Placed in series with signal lines, each two-terminal device guards sensitive circuit components against voltages near and beyond the normal supply voltages. They are used at interfaces where sensitive circuits are connected to the external world and could encounter damaging voltages (up to 35 V beyond the supply rails) during power-up, power-down, or fault conditions.

The MAX366 contains three independent protectors and the MAX367 contains eight. They can protect analogue signals using either unipolar (4.5–36 V) or bipolar (±2.25 to ±18 V) power supplies. Each protector is symmetrical. Input and output terminals may be freely interchanged.

The devices are voltage-sensitive MOSFET transistor arrays that are normally on when power is applied and normally open circuit when power is off. With ±10 V supplies, on-resistance is 100 Ω maximum and leakage is less than 1 nA at +25 °C.

When signal voltages exceed or are within about 1.5 V of either power-supply voltage (including when power is off), the two-terminal resistance increases greatly, limiting fault current as well as output voltage to sensitive circuits. The protected side of the switch maintains the correct polarity and clamps about 1.5 V below the supply rail. There are no 'glitches' or polarity reversals going into, or coming out of, a fault condition.

Background information

When a voltage outside the supply range is applied to most integrated circuits, there is a strong possibility they will be damaged or 'latch up' (that is, fail to operate properly even after the offending voltage is removed). If an IC's input or output pin is supplied with a voltage when the IC's power is off, and power is subsequently applied, the device may act as an SCR and destroy itself and/or other circuitry. Such 'faults' are commonly

ELECTRICAL CHARACTERISTICS

(V+ = +15V, V- = -15V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	TEMP. RANGE	MIN	TYP	MAX	UNITS
Analog Signal Range	V_{IN}, V_{OUT}	(Note 1)	All	(V+ - 40)		(V- + 40)	V
Fault-Free Analog Signal Range	V_{IN}, V_{OUT}	V+ = 15V, V- = -15V (Note 2)	All	-11		11	V
Analog-Signal Output Range (Fault)	V_{OUT}	V_{IN} = V+ or V-, 100kΩ < R_{OUT} < 1000MΩ (Note 1)	All	(V- + 3)		(V+ - 1.5)	V
Analog-Signal-Path Resistance	$R_{(IN-OUT)}$	V+ = 15V, V- = -15V, V_{IN} = ±10V, I_{OUT} = 1mA	+25°C		62	85	Ω
			C, E			100	
			M			125	
		V+ = 10V, V- = -10V, V_{IN} = ±5V, I_{OUT} = 1mA	+25°C		62	100	
			C, E			125	
			M			150	
		V+ = 5V, V- = -5V, V_{IN} = ±2V, I_{OUT} = 1mA	+25°C		140	350	
			C, E, M			400	
Signal-Path Resistance Match	$\Delta R_{(IN-OUT)}$	V_{IN} = ±10V, I_{OUT} = 1mA	+25°C			7	Ω
			C, E, M			10	
Signal-Path Leakage (Power Off)	$I_{IN(OFF)}$	V+ = V- = 0V, V_{IN} = ±35V, V_{OUT} = open circuit	+25°C	-10		10	nA
			C, E, M	-1000		1000	
Signal-Path Leakage (without Fault Condition)	$I_{OUT(ON)}$	V_{IN} = V_{OUT} = ±10V	+25°C	-1		1	nA
			C, E, M	-100		100	
Signal-Path Leakage (with Fault Condition)	$I_{IN(ON)}$	V_{IN} = ±25V, V_{OUT} = open circuit	+25°C	-10		10	nA
			C, E, M	-1000		1000	
Signal-Path Leakage (with Overvoltage)	$I_{IN(OFF)}$	V+ = V- = 0V, V_{OUT} = 0V, V_{IN} = ±35V	+25°C	-10		10	nA
			C, E, M	-1000		1000	
POWER SUPPLY							
Power-Supply Range	V+, V-		+25°C, C, E, M	0		±18	V
Power-Supply Range (without Fault Condition)	V+, V-	$R_{(IN-OUT)}$ < 1000Ω (Note 2)	+25°C, C, E, M	±2.25		±18	V
Power-Supply Current	I+, I-		+25°C	-1		1	µA
			C, E, M	-10		10	

Note 1: Guaranteed, but not tested.

950116 - T1

MAXIM

Table 1. Electrical characteristics.

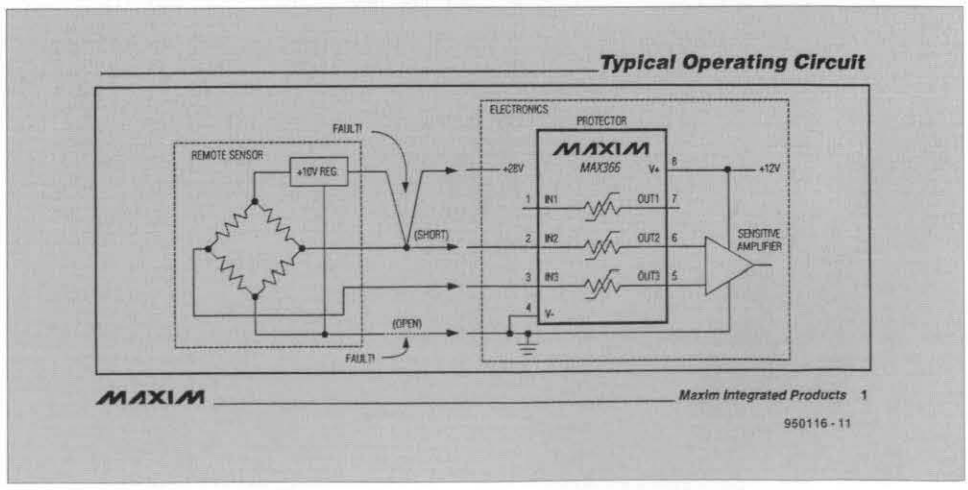


Fig. 1. Typical operating circuit.

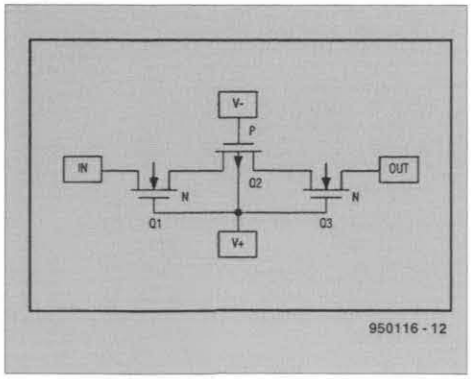


Fig. 2. Simplified internal structure.

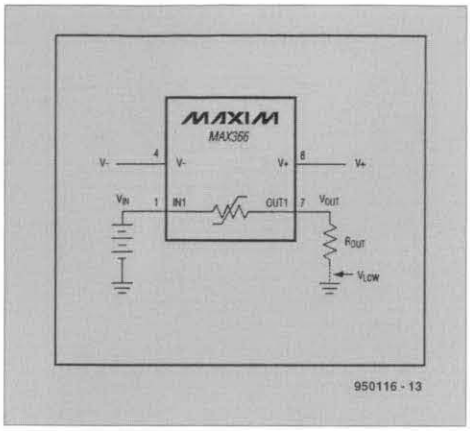


Fig. 3. Application circuit.

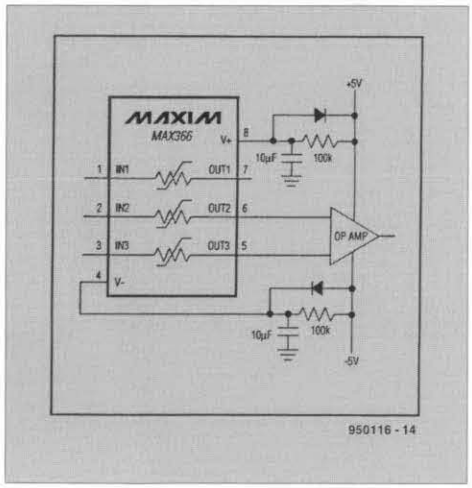


Fig. 4. Turn-on delay.

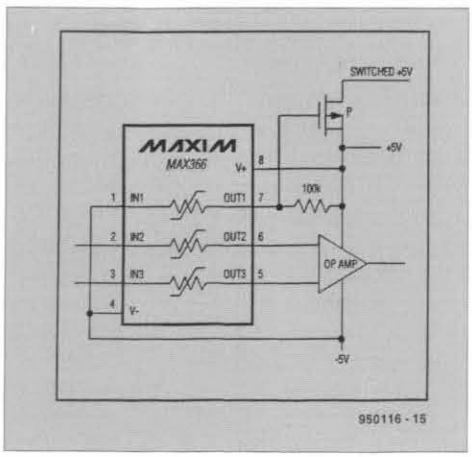


Fig. 5. Power-supply sequencing.

encountered in modular control systems where power and signals to interconnected modules may be interrupted and re-established at random. They can happen during production testing, maintenance, start-up, or power 'brownout'.

The MAX366/MAX367 are designed to protect delicate input and output circuitry from overvoltage faults up to ±40 V (with or without power applied), in devices such as op amps, analogue-to-digital converters (ADCs), digital-to-analogue converters (DACs), and voltage references. These circuit protectors automatically limit signal voltages and currents to safe levels without degrading normal signal performance, even in very-high-imped-ance circuits. They are powered by the power supply of the protected circuit and inserted into the signal lines. There are no control lines, programming pins, or adjustments.

Unlike shunt diode networks, these devices are low-impedance FETs that become high impedance during a fault condition, so fault current and power dissipation are extremely low. Equally important, leakage current during normal and fault conditions is extremely low. In addition, unlike most discrete networks, these parts protect circuits both when power is off and during power transitions.

Detailed description

Figure 2 shows the simplified internal construction of each protector inside the MAX366/MAX367. Each circuit consists of two n-channel FETs and one p-channel FET. All the FETs are enhancement types; that is, the n-channels must have about 1.3 V of positive gate voltage in order to conduct, and the p-channel must have around 2 V of negative gate voltage in order to conduct.

During normal operation, V^+ is connected to a positive potential and V^- is connected to a negative potential. Since their gates are tied to V^+, transistors Q_1 and Q_3 conduct as long as their sources are at least 1.3 V below V^+ (the n-channel gate threshold). Transistor Q_2's gate is tied to V^-, so it conducts as long as its source is 2 V or more above V^- (the p-channel gate threshold).

As long as the signal is within these limits, all three transistors conduct and a low-resistance path is maintained from the IN to OUT pin. (Note that, since the device is symmetrical, IN and OUT pins can be interchanged) When the signal is beyond the gate threshold of either Q_2 or Q_1/Q_3, the path resistance rises appreciably. When power is off, none of the transistors have gate bias, so the circuit from IN to OUT is open.

Normal operation

In normal operation, the protector is placed in series with the signal line and the power supplies are connected to V^+ and V^- (see **Fig. 3**). V^- is ground when operation is from a single supply. When power is applied, each protector acts as a resistor in the signal path. Any voltage source on the 'input' side of the switch will be conducted through the protector to the output. (Note that, since the protector is symmetrical, IN and OUT pins can be interchanged).

If the output load is resistive, it will draw current, and a voltage divider will be formed with the internal resistance, so the output voltage will be lower than the input voltage. Since the internal resistance is typically smaller than 100 Ω, high-impedance loads will be relatively unaffected by the presence of the protector. The protector's path resistance is a function of the supply voltage and the signal voltage.

Power off

When power is off, that is, $V^+ = V^- = 0$ V, the protector is a virtual open circuit, and all voltages on each side are isolated from each other up to ±40 V. With ±40 V applied to the input pin, the output pin will be 0 V, regardless of its resistance to ground.

Fault conditions

A fault condition exists when the voltage on either signal pin is within 1.5 V of either supply rail or exceeds either supply rail. This definition is valid when power is applied and when it is off, as well as during all the states as power ramps up or down.

During a fault, the protector acts as a variable resistor, conducting only enough to sustain the other side of the switch within about 1.5 V of the supply rail. This fault is know as the 'fault knee voltage' and is not symmetrical. It is about 1.3 V down from the positive supply (V^+ pin) or around 2.0 V up from the negative supply (V^- pin). Each fault knee voltage varies slightly with supply voltage, with output current, and from device to device.

During a fault condition, all the fault current flows from one signal pin through the protector and out the other signal pin. No fault current flows through either supply pin. (There will be a few pico-amps of leakage current from each signal pin to each supply pin, but this is independent of fault current).

During the fault condition, enough current will flow to maintain the output voltage at the fault knee voltage, so the fault current is a function of the output resistance and the supply voltage. **The output voltage and current have the same polarity as the fault**.

The maximum input fault voltage is 40 V from the 'opposite-polarity supply rail'. This means the input can go to ±35 V with ±5 V supplies or to ±25 V with ±15 V supplies. The fault voltage is highest (±40 V) when the supplies are off ($V^+ = V^- = 0$ V).

Using the circuit of Fig. 3, the approximate fault currents are:

1. for positive faults:
$$I_F \approx (V^+ - 1.3 - V_{LOW}) : R_{OUT};$$

2. for negative faults:
$$I_F \approx (V^- + 2 + V_{LOW}) : R_{OUT},$$

where V_{LOW} is the terminating voltage at the far end of R_{OUT}. $V_{LOW} = 0$ V when R_{OUT} is grounded.

The current through each protector should never exceed 30 mA. Always calculate the power dissipated by all the protectors in worst-case conditions (maximum voltage and current through each protector) to ensure the package dissipation limit is not reached.

With single-supply operation, grounded loads will have zero voltage (and current) whenever the input voltage is below about 2 V. In effect, both the IN and OUT pins are in fault condition.

A special case arises when the power is off: the part is in a perpetual fault condition but no fault current flows because all the internal FETs are off.

Single-supply output operation

Single-supply operation is a special case. Signals can not go to ground, since from 0 V to about +2 V is a fault condition.

Very-low-current operation

When the protector's output side is connected to very high resistance, very-low-current loads (such as op amp inputs), a small leakage current flows from the input to the output during fault conditions. This current is typically below a nano-ampere, but, if the output resistance is high enough, it can cause the output voltage to exceed the supply voltages during fault conditions.

This condition can be self-correcting, however, if the high-resistance load has protection diodes to the supply rails (either external or internal to the op amp). These diodes conduct the leakage current to the supply rails and safely limit the output voltage. An alternative is to add a high-value resistor to ground in parallel with the load. This resistor may be as low as 1000 MΩ; its value must be determined experimentally at the highest expected operational temperature.

The fault protectors will not normally be used with high-impedance FET input amplifiers that lack input protection diodes. Such amplifiers are fragile and are normally reserved for use when ultra-low leakage (pA) is needed. The MAX366/MAX367 have nano-amperes of leakage, which would negate the low leakage of the unprotected amplifier.

Low-voltage operation

The MAX366/MAX367 'operate' with supply voltages all the way down to 0 V, but what they do to the signal is not obvious. With a total supply voltage of 3.5 V, the protector is in a fault condition with nearly any input that is not close to 2 V. Below 3.5 V (including power off), the protector is perpetually in a fault condition, that is, high impedance.

When the supply voltage(s) ramps up (and/or down) from zero, the signal path is initially in a fault condition (open), until the supply voltage passes the input voltage. The output starts at zero and is delayed from reaching the input voltage as the part comes out of the fault condition. If the supply voltage exceeds about 3.5 V, but never exceeds the input voltage, the output will follow the supply, always remaining about 1.3 V below the positive supply voltage or 2 V above the negative supply voltage. If the input voltage subsequently comes out of the fault condition, the output returns to the input value. This set of conditions is exactly reversed when power ramps down to zero.

Since the input and output pins are identical and interchangeable, predicting whether or not the part is in a fault condition is easy: if either IN or OUT exceeds V^+ or V^-, a fault condition exists and the current that flows will be just enough to cause the other signal pin (OUT or IN) to approach the appropriate supply rail.

Typical applications

Driven switches

The MAX366/MAX367 have low supply currents (<1 µA), which allows the supply pins to be driven directly by other active circuitry, instead of connected directly to the power sources. In this configuration, the parts can be used as driven fault-protected switches with V^+ or V^- pins used as the control pins. For example, if the V^- pin is grounded, you can turn the V^+ pin on and off by driving it with the output of a CMOS gate. This effectively connects and disconnects three or eight separate signal lines at once. (If bipolar signals, or signals that go to ground, are being switched, the V^- pin must be driven simultaneously to a negative potential. Always ensure that the driving source(s) does not drive the V^+ pin more negative than the V^- pin.

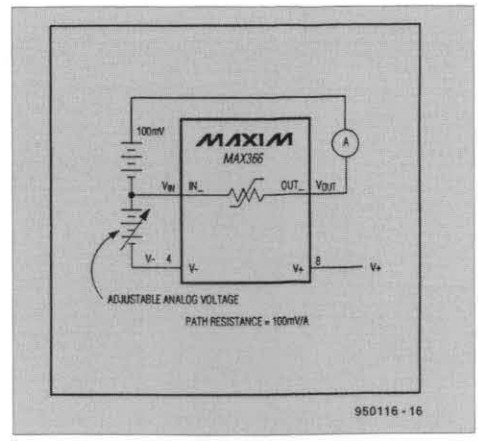


Fig. 6. Path-resistance measuring circuit.

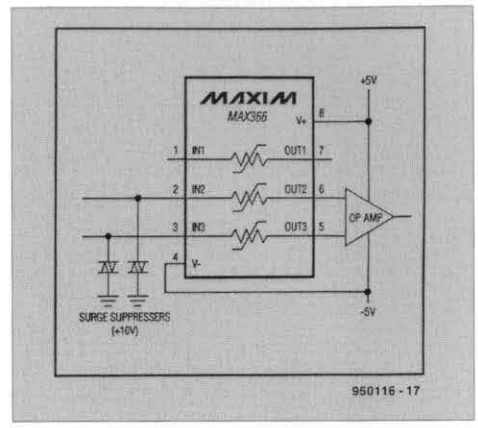


Fig. 7. Surge-suppression circuit.

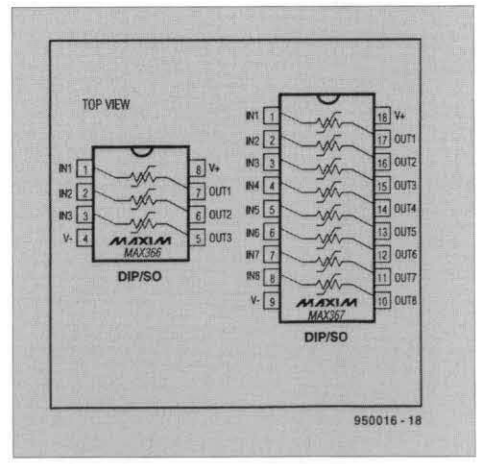


Fig. 8. Pin configurations.

Figure 4 shows a simple turn-on delay that takes advantage of the MAX366's low power consumption. The two RC networks cause gradual application of power to the MAX366, which in turn applies the input signals smoothly after the amplifier has been stabilized. The two diodes discharge the two capacitors rapidly when power is turned off.

This circuit can be tailored to nearly any rate of turn-on by selecting the RC time constants in the V^+ and V^- pins, without affecting the time constant of the measuring circuit.

Protectors as circuit elements

Any of the individual protectors in a MAX366 or MAX367 may be used as a switched resistor, independent of the functions of other elements in the same package. For example, **Fig. 5** shows a MAX366 with two of the protectors used to protect the input of an op amp, and the third element used to sequence a power supply. Combining the circuits of Fig. 4 and Fig. 5 produces a delayed action on the switched +5 V, as well as smooth application of signals to the amplifier input.

Testing circuit procedures

Measuring path resistance

Measuring path resistance requires special techniques, since path resistance varies greatly with the IN and OUT voltages relative to the supply voltages. Conventional ohmmeters should not be used, for two reasons: 1) the applied voltage and currents are usually not predictable, and 2) the true resistance is a function of the applied voltage, which is altered by the ohmmeter itself. Autoranging ohmmeters are particularly unreliable.

Figure 6 shows a circuit that can give reliable results. This circuit uses a 100 mV voltage source and a low-voltage-drop ammeter as the measuring circuit, and an adjustable supply to sweep the analogue voltage across its whole range. The ammeter must have a voltage drop of not more than 1 mV (at any current) for accurate results. (A Keithley Model 617 Electrometer has a suitable ammeter, appropriate ranges, and a built-in voltage source designed for this type of measurement). Measurements are made by setting the analogue voltage, measuring the current, and calculating the path resistance. The procedure is repeated at each analogue voltage and supply voltage.

It is important to use a voltage source of 100 mV or less. As shown in Fig. 4, this voltage is added to the V_{IN} voltage to form the V_{OUT} voltage. Using a higher voltage could cause the OUT pin to go into a fault condition prematurely.

High-frequency performance

In 50 Ω systems, signal response is reasonably flat up to several megahertz. Above 5 MHz, the response has several minor peaks, which are highly layout dependent. Because the path resistance is dependent on the supply voltage and signal amplitude, the impedance is not controlled. Adjacent channel attenuation up to 5 MHz is about 3 dB above that of a bare IC socket, and is caused entirely by capacitive coupling.

Pulse response is reasonable, but because the impedance changes rapidly, fast rise times may induce ringing as the signal approaches the fault voltage. At very high amplitude (such as noise spikes), the capacitive coupling across the signal pins will transfer considerable energy, despite the fact thet the d.c. path is a virtual open circuit.

High-voltage surge suppression

These devices are not high-voltage arresters, nor are they substitutes for surge suppressors. In systems that use these forms of protection, however, the MAX366/MAX367 can fill a vital gap. **Figure 7** shows a typical circuit. Although the surge suppressors are extremely fast shunt elements, they have very soft current knees. Their clamp voltage must be chosen well above the normal signal levels, because they have excessive leakage currents as the knee is approached. This current can interfere with normal operation when signal levels are low or impedances are high. If the clamp voltage is too high, however, the input can be damaged. [950116]

CODING AND MODULATION TECHNIQUES IN ERMES

Brian P. McArdle

1. Introduction

The term ERMES stands for European Radio Message System. It was introduced by the European Telecommunication Standards Institute (ETSI) as a pan-European paging system. Both the equipment specifications and frequency band are harmonized. The only difference between countries is an identifier within the transmissions. The purpose of this artcile is to examine the technical characteristics – especially the coding and modulation operations. Any comments are purely personal.

2. General arrangement

Figure 1 illustrates the structure of an ERMES system. There is one PNC which manages an entire network. A particular area within a network is managed by a PAC which controls all base stations in that area. These two stages could be described as the telecommunications part of a network. Radiofrequency transmissions to pagers come from the base stations where the coding and modulation techniques in ERMES are implemented. Pagers are receive only and have no facility for transmission. These two sections form the radio part of the system. For efficient operation and to minimize the possibility of harmful interference to receivers, a network is divided such that transmissions into overlapping or adjacent areas fom different base stations are separated with respect to time or frequency. This point does not require further explanation as it refers to network planning. It suffices to note that the normal spectrum engineering parameters must be followed in the locations of base stations – even where two such stations are each under a different PAC. However, for the purposes of this article it is assumed that an ERMES system is fully operational.

Transmissions by a base station are in batches of 16 (sub-sequence) with a duration of 12 seconds. A cycle consists of five sub-sequences with a duration of one minute. A full sequence lasts 60 minutes. Figure 2 shows the procedure. (In this respect, ERMES is quite slow in comparison to other recently developed radio system, such as GSM and TETRA). A batch number within a sub-sequence is particularly important. During operation, a pager is assigned to one of the 16 batches in accordance with an identification code which is preprogrammed into the unit. Consequently, the receiver population is effecively divided into 16 groups.

From the radiofrequency viewpoint there is a total of 16 channels from 169.425 MHz to 169.8 MHz in steps of 25 kHz. The batches are arranged such that each batch number occurs at a different time period on each radiofrequency channel. A unit is programmed to monitor its own batch on the full 16 or a selected number of channels. If it is addressed on a particular channel, it remains on that channel for the duration of the message. However, it must be addressed initially in the appropriate batch by the base station.

The paging receivers must operate in on of the following categories.

(a) Tone only where a message is simply a sequence of tones (e.g., alert signal.
(b) Numeric where a message consists of numeric data and tones.
(c) Alphanumeric where a message consists of letters and numbers. A unit must also be capable of processing the other two categories.
(d) Transparent data that does not have a specific form. A unit must also be capable of operating in the other three categories.

The coding methods and modulation techniques are discussed in the next two sections.

3. Message structure

The format of a transmission from a base station is illustrated in Fig. 3. There are four sections. Number 1 consists of a preamble (30 bits) and synchronization word (30 bits). The

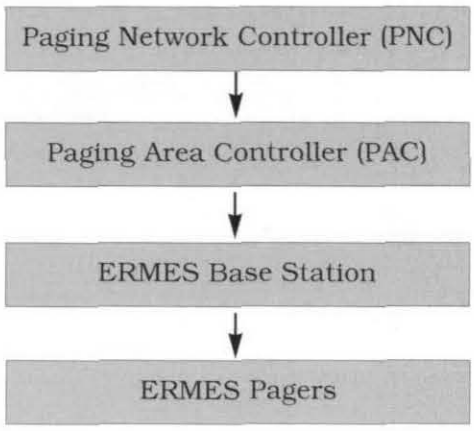


Fig. 1. ERMES system. The pagers are receive only.

SYSTEM INFORMATION contains items such as country code (7 bits), cycle number (6 bits) and batch number (4 bits). Further details are not really necessary. It suffices to state that both blocks have fixed sizes. The lengths do not vary during operation.

The ADDRESS PARTITION contains an integral number of addresses in descending order. The number is variable, but the maximum value is 140. The section ends with a terminator (30 bits) which can be recognized by pagers. During operation, a pager monitors for its own particular address. If it is addressed, the data follows in the same or successive batches.

The MESSAGE PARTITION contains the actual data. Different messages are separated by delimiters. Each message starts with a header of fixed length (36 bits) and is followed by data and check bits. In an ERMES base station, the encoding procedure consists of two main steps. The original data is formed into codewords of 18 bits. A further 12 bits are produced by a cyclic code (Appendix B) such that each word becomes 30 bits. For easy

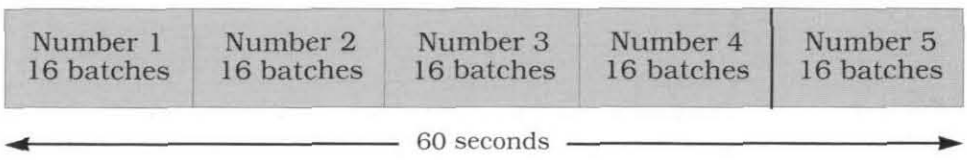


Fig. 2. Cycle consisting of 5 sub-sequences. Each sub-sequence has a duration of 12 seconds and contains 16 batches.

#1 Synchronization partition	#2 System information partition	#3 Address partition	#4 Message partition

Fig. 3. MESSAGE STRUCTURE in ERMES. A message is divided into four sections. Numbers 1 and 2 have fixed lengths.

implementation, the standard lists a binary matrix (it is not necessary to undertake complex mathematical calculations and the derivation of the matrix need not be analysed). The operation is simply the usual form of multiplication for matrices. The input is written as a row vector (1 row and 18 columns). The encoding matrix has 18 rows and 30 columns and is fixed for all operations. Consequently, the result is a row vector (1 row and 30 columns as follows:

[Data (1, 18)] [Encoding matrix (18, 30)] = [Codeword (1, 30)].

This is the most complicated mathematical operation in the entire process. The next and final step is merely a re-ordering in the position of bits. It is an interleaving operation over nine codewords and its purpose is to reduce the effect of erroneous bits at the decoder stage. It is a simple and effective process that does not involve complex mathematical functions. The nine codewords are grouped such that the 30th bit in each word is input to the modulator. This is followed by the 29th bit and so on. Thus, the MSB is transmitted first and is followed by the other bits down to the LSB. In the decoder the bits are simply restored to their original positions. At first glance, the operation may appear unnecessary since it adds another layer to the encoding procedure. However, a number of errors occurring together during transmission would not remain as a group to disrupt a receiver. The inverse interleaving operation would scatter the erroneous bits which would be detected by the cyclic decoder (Appendix B).

The foregoing summary of the encoding operation is rather simplified but it does illustrate the main points. The reader is referred to the specification for the exact procedure. The modulation technique is disccused in the next section.

An important point to note about ERMES is that it is not a confidential system. There is no encryption operation within the specification for the protection of data. (GSM cellular telephone and TETRA in mobile radio have secrecy operations as part of the equipment standards.) A hacker with the correct receiving equipment could listen in to messages without any difficulty. The positive side is that messages tend to be short and would not contain sensitive information.

4. Modulation

The method is described in the specification as 4PAM/FM. Following the encoding operation, the final block is input to the modulator – 2 bits at a time. Figure 4 shows the technique. Each block of 2 bits produces a pulse-amplitude modulated signal. There is a total of four levels corresponding to the various bit combinations. After Stage 1 the signal is input to a filter (Stage 2) with a specific response which need not be examined in an overview. This becoms the equivalent of the modulation symbol for the final step (Stage 3). The result is a particular frequency shift such that the instantaneous frequency is one of four values (Appendix A). The modulation rate is 3125 baud.

There have been compatibility problems between ERMES and services in adjacent bands. There are PMR and broadcast (Band 3 Television) services in the bands immediately below and above ERMES. For example, in the case of ERMES and PMR base station receivers, it was estimated that isolation of up to 90 dB could be required in densely populated areas in order to avoid interference due to selectivity, spurious responses and blocking [2]. Reception of Band 3 Television was also affected – especially at the edge of the band. However, these effects are due to the particular frequency band rather than the modulation technique. They can be eliminated only through an analysis of the spectrum engineering requirements.

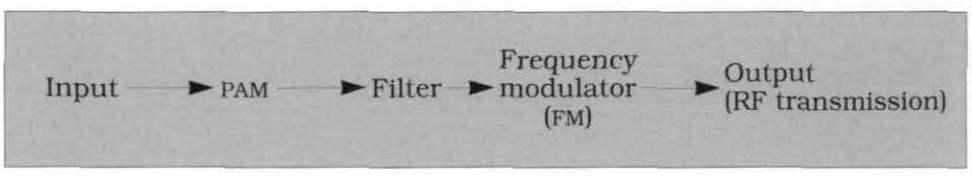


Fig. 4. MODULATION TECHNIQUE in ERMES.

5. Remarks

ERMES is a good combination of radio and telecommunications services. There is good coverage at VHF and units are small and compact. It has the advantage of being a harmonized system which is intended for operation throughout the European Union. There are no modifications by individual countries. A unit does not have to be designd to meet separate national standards. While a user does not have the benefit of two-way communication, the system can deliver messages quickly with a high degree of reliability.

6. References

[1] ETSI Standard ETS 300-133.

[2] ETSI Standard ETS 300-086: 'Technical characteristics and test conditions for radio equipment with an external antenna connector, intended primarily for analogue speech, for use in the land mobile service.'

[3] *Radio Spectrum Management*
D.J. Withers
Peter Peregrinus Ltd
United Kingdom (1991).

[4] *Error Control Techniques for Digital Communications*
Arnold M. Michelson & Allen H. Levesque
John Wiley & Sons (1985)

7. Appendix A

$$f_i = f_c + \Delta f$$

where f_i = instantaneous frequency
f_c = carrier frequency
Δf = frequency shift

Frequency shift (Hz)	Symbol
+4687.5	10
+1562.5	11
–1562.5	01
–4687.5	00

The table has the form of a Gray Code such that successive shifts differ in just one bit. The filter is not analysed, but has a response similar to that of a 10th order low-pass Bessel. The data rate is 6250 bits/second which is twice the symbol rate (Section 4).

8. Appendix B: Generator polynomial

The generator polynomial for the (30, 18) cyclic code is

$(x^{12} + x^{11} + x^9 + x^7 + x^6 + x^3 + x^2 + 1)$

and produces 12 check bits from the original data bits using the standard technique. The Minimum Distance is 6 and the code is capable of correcting two errors per block. The significant point is that arithmetical operations are modulo 2. The only coefficients are either '1' or '0'. Any mathematical operations must preserve this representation. The procedure is followed by an interleaving process where the bits are re-ordered.

9. Appendix C: Paging codes

Before the development of ERMES, two other paging codes were widely used: POCSAG and GOLAY.

POCSAG uses a codeword of 32 bits with the format shown in the box at the bottom of this page.

The ten check digits are generated in the usual manner by the polynomial

$(x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1)$

which is a factor over GF(2) of $(x^{31} + 1)$. In this type of article the exact mathematical properties do not require an in-depth analysis. The parity bit is added at the end to allow for the detection of an odd number of errors. The final length is 32 bits. For further information, the reader is referred to Ref. [4].

GOLAY uses a codeword of 23 bits. The check bits are generated by a (23, 12) cyclic code by the standard technique. Both GOLAY and POCSAG are the same in this regard. The generator polynomial can be either

$(x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1)$
or
$(x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1)$

which are factors over GF(2) of $(x + 1)$. They generate 11 check bits resulting in a block of 23 bits. The code can detect up to three errors per block. Unfortunately, the main properties of this code are unique.

[952007]

Bit number	32 31 12	11 10 2	1
Function	Data	Check	Parity

CORRECTION

'Coding & Modulation Techniques in ERMES' (October 1995)

The seventh line before the end of the article (page 75) reads:

which are factors over GF(2) of $(x + 1)$.

This should read:

which are factors over GF(2) of $(x^{23} + 1)$.